



Universidad de Valladolid

E.T.S. de Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática
Mención Ingeniería del Software

Geolocalization Game: app Android para la generación de
juegos geográficos a medida

Autor:

Maykel Alberto García Olivares

Tutora:

Yania Crespo González Carvajal

Agradecimientos

Quisiera aprovechar estas líneas para agradecer a todas aquellas personas que de una manera u otra han hecho posible mi formación como ingeniero.

En especial me gustaría agradecer a mi madre Dania, a mi padre Onel y a mi segundo padre José Orlando por siempre apoyarme cuando he atravesado los momentos más difíciles durante la carrera , tanto cuando han estado cerca de mí, como cuando hemos estado separados.

Agradecer también a mis amigos Gabriel, Lorena, Luis, Sadiel, Raidel, Ramón, Víctor, Diego... gracias a todos ustedes siempre pude sacar una sonrisa cuando más lo necesitaba.

La vida con el apoyo de la familia (Michel, Irene, Roisy, mis primos, mis suegros, mis tíos, mis sobrinos y sobrinas) siempre es más fácil, gracias a ustedes por estar ahí y siempre poder contar con vosotros.

Nunca olvidaré a todos los que estuvieron junto a mi apoyándome incondicionalmente y hoy ya no están, a ustedes, un agradecimiento muy especial.

Por último, agradecer a mi tutora Yania por confiar en mí para la realización de este proyecto y por todo el apoyo recibido durante las distintas fases de desarrollo. Quisiera agradecerle también todo el apoyo que siempre me ha brindado durante mi estancia académica en la UVA.

Tabla de contenido

Introducción	1
1.1. Contexto y Motivación	1
1.2. Objetivos	2
1.3. Resumen del contenido de la memoria	3
Entorno Tecnológico	5
2.1 Tecnología para el uso de mapas en Android	5
2.1.1. Necesidad del uso de mapas	5
2.1.2 Algunas Tecnologías disponibles	5
2.2. Tecnologías utilizadas	6
2.2.1. Google Maps API v3:	6
2.2.2. HTML5:	6
2.2.3. CSS	7
2.2.4. Java Script	7
2.2.5. Java	7
2.2.6. SQLite	7
Desarrollo de una aplicación en Android	9
3.1. El sistema operativo Android	9
3.1.1. Introducción	9
3.1.2. Breve Historia	9
3.1.3. Características y especificaciones actuales	10
3.1.4. Nombre de las versiones y fragmentación	12
3.1.1. Arquitectura de Android	14
3.2. Las Aplicaciones	15
3.2.1. Introducción	15
3.2.2. Herramientas para el desarrollo de Android	15
3.2.3. Componentes	16
3.2.3.1. Las Activity	16
3.2.3.2. Los Service	19
3.2.3.3. ContentProvider	19
3.2.3.4. BroadcastReceiver	19
3.3. Estructura de un proyecto en Android	20
3.3.1. Introducción	20

Tabla de Contenido

3.3.2.	Estructura.....	20
3.3.3.	El fichero AndroidManifest.xml.....	22
Gestión y planificación del proyecto		25
4.1.	Introducción.....	25
4.2.	Gestión de riesgos	26
4.2.1.	Identificación y análisis de riesgo.....	27
4.2.2.	Plan de riesgos	28
4.2.3.	Supervisión y seguimiento de riesgos.....	31
4.3.	Planificación.....	31
4.3.1.	Recursos	31
4.3.2.	Estimación temporal	32
4.3.3.	Distribución temporal	33
4.3.4.	Costes	36
Requisitos		37
5.1.	Requisitos Funcionales.....	37
5.2.	Requisitos No Funcionales	38
5.3.	Requisitos de Información.....	39
5.4.	Casos de Uso	40
Análisis.....		45
6.1.	Modelo del Dominio.....	45
6.1.	Diagrama de Casos de Uso.....	46
6.1.	Diagrama de Secuencias.....	47
6.1.	Diagrama de Clases	51
Diseño.....		53
7.1.	Estructura de Diseño.....	53
7.2.	Descripción de la arquitectura	53
7.2.1.	MVC.....	53
7.2.2.	Arquitectura global del sistema	54
7.3.	Patrones de diseños utilizados	56
7.3.1.	Patrón Singleton	56
7.3.2.	Iterator	57
7.1.	Diagrama de clases por funcionalidad.....	57
7.1.1.	Jugar	58
7.1.1.	Crear Juego	59
7.1.1.	Editar un Juego	60

7.1.1.	Importar Juego	61
7.1.1.	Exportar Juego	62
7.1.2.	Ayuda.....	62
7.1.1.	Relación entre las Activities.....	63
7.2.	Modelo de datos.....	63
7.2.1.	Base de datos SQLite	63
7.2.2.	Modelo de datos de la App.....	64
7.1.	Diagramas de secuencias	65
7.1.1.	Exportar un Juego	65
7.1.2.	Mostrar Ayuda	66
7.1.1.	Crear un Juego, añadir una localidad	66
7.2.	Diagrama de Despliegue	67
Implementación.....		69
8.1.	Introducción	69
8.2.	Utilización de la API v3 de Google Maps en Android	69
8.2.1.	WebView	69
8.2.1.	Creando la página HTML que contendrá el mapa.....	71
8.2.1.	Cargando el mapa en Android.....	74
8.3.	Vinculando código de JavaScript a Android.....	74
8.1.	Ejecutando una función JavaScript declarada en la página HTML desde Android ...	75
Pruebas.....		77
9.1.	Introducción	77
9.2.	Plan de pruebas	78
9.2.1.	Pruebas unitarias	78
9.2.2.	Pruebas de integración	78
9.2.3.	Pruebas de sistema	78
9.3.	Casos de Pruebas	78
9.3.1.	Creación de un juego nuevo	79
9.3.1.	Jugar.....	80
9.3.1.	Editar.....	81
9.3.2.	Importar Juego Actual.....	82
9.3.1.	Exportar Juego Actual.....	83
9.3.2.	Ayuda.....	83
9.3.3.	Elegir Filtro	83
9.3.4.	Modo de Juego: ¿Cómo se llama este punto?	84

Tabla de Contenido

9.3.5.	Niveles de juego: Intermedio, Avanzado	84
9.3.1.	Conexión a internet	85
9.4.	Resultados de las pruebas realizadas	85
9.4.1.	Pruebas en Móvil Samsung Galaxy S3	86
9.4.2.	Pruebas en Samsung Galaxy Tab 3 de 7 pulgadas	87
9.4.1.	Fallos encontrados	88
	Conclusiones.....	89
10.1.	Objetivos Cumplidos.....	89
10.2.	Valoración Personal	90
10.3.	Trabajo Futuro.....	90
	Bibliografía.....	91
	Libros.....	91
	Sitios web	91
	Blog	93
	Apéndice A: Manual de Usuario	95
	Instalación de Geolocalization Game	95
	Primera Vez	95
	Crear un Juego	96
	Jugar	97
	Editar un Juego	98
	Importar Juego Actual	99
	Exportar Juego Actual	99
	Ayuda	100
	Opciones	101
	Apéndice B: Glosario de términos.....	103
	Apéndice C: Contenido del CD	105

Índice de Figuras

FIGURA 1: ORIGEN Y DESARROLLO	10
FIGURA 2: RECOLECCIÓN DE DATOS ENTRE DICIEMBRE DE 2014 Y FEBRERO DE 2015.....	13
FIGURA 3: RECOLECCIÓN DE DATOS ENTRE DICIEMBRE DE 2014 Y FEBRERO DE 2015.....	13
FIGURA 4: ARQUITECTURA DE ANDROID.....	14
FIGURA 5: EL CICLO DE VIDA DE UNA ACTIVIDAD.....	18
FIGURA 6: ESTRUCTURA DE UN PROYECTO ANDROID EN ECLIPSE	22
FIGURA 7: FICHERO ANDROIDMANIFEST.XML	23
FIGURA 8: PROCESO UNIFICADO.....	26
FIGURA 9: DIAGRAMA DE GANNT DE LA CALENDARIZACIÓN ESTIMADA	34
FIGURA 10: MODELO VISTA CONTROLADOR.....	54
FIGURA 11: ARQUITECTURA DE LA APLICACIÓN.....	55
FIGURA 12: DIAGRAMA OMT DE SINGLETON	56
FIGURA 13: ESTRUCTURA ITERATOR	57
FIGURA 14: DIAGRAMA DE CLASES JUGAR	58
FIGURA 15: DIAGRAMA DE CLASES CREAR JUEGO.....	59
FIGURA 16: DIAGRAMA DE CLASES EDITAR JUEGO.....	60
FIGURA 17: DIAGRAMA DE CLASES IMPORTAR UN JUEGO	61
FIGURA 18: DIAGRAMA DE CLASES EXPORTAR UN JUEGO.....	62
FIGURA 19: DIAGRAMA DE CLASES AYUDA	62
FIGURA 20: DIAGRAMA DE RELACIÓN ENTRE ACTIVITIES	63
FIGURA 21: BASE DE DATOS SQLITE.....	64
FIGURA 22: TABLA DE LA BASE DE DATOS	65
FIGURA 23: DIAGRAMA DE SECUENCIA IMPORTAR JUEGO.....	65
FIGURA 24: DIAGRAMA DE SECUENCIA MOSTRAR AYUDA.....	66
FIGURA 25: DIAGRAMA DE SECUENCIA CREAR JUEGO, AÑADIR UNA LOCALIDAD	66
FIGURA 26: DIAGRAMA DE DESPLIEGUE	67
FIGURA 27: PANTALLA DE MENÚ PRINCIPAL DE LA PRIMERA INSTALACIÓN.....	95
FIGURA 28: CREAR JUEGO.....	96
FIGURA 29: JUGAR	97
FIGURA 30: EDITAR JUEGO.....	98
FIGURA 31: IMPORTAR JUEGO	99
FIGURA 32: EXPORTAR JUEGO	100
FIGURA 33: AYUDA.....	100
FIGURA 34: OPCIONES	101
FIGURA 35: ELEGIR FILTRO FIGURA 36: MODO DE JUEGO FIGURA 37: DIFICULTAD DEL JUEGO	102

Índice de Tablas

TABLA 1: PRINCIPALES CARACTERÍSTICAS DE ANDROID	11
TABLA 2: IDENTIFICACIÓN DE RIESGOS	27
TABLA 3: R01- SE NECESITA MÁS TIEMPO PARA EL ESTUDIO DE LA TECNOLOGÍA A USAR.....	28
TABLA 4: R02- FALTA DE MOTIVACIÓN.	28
TABLA 5: R03- EXAMEN EXTRAORDINARIO DE FÍSICA.	28
TABLA 6: R04- REALIZACIÓN DE LAS PRÁCTICAS DE EMPRESA.	29
TABLA 7: R05- CAMBIOS DE LOS REQUISITOS FUNCIONALES UNA VEZ EMPEZADO EL PROYECTO.....	29
TABLA 8: R06-PLANIFICACIÓN MUY OPTIMISTA.	29
TABLA 9: R07- FALLOS AL PROBAR LA APLICACIÓN EN DISPOSITIVOS CON CARACTERÍSTICAS DIFERENTES.	30
TABLA 10: R08- DESARROLLO DE INTERFACES INCORRECTAS.	30
TABLA 11: SUPERVISIÓN Y SEGUIMIENTO DE RIESGOS.	31
TABLA 12: DURACIÓN ESTIMADA DE LAS TAREAS.....	33
TABLA 13: DURACIÓN REAL DE LAS TAREAS.....	34
TABLA 14: HORAS INCREMENTADAS DE LAS TAREAS.....	35
TABLA 15: COSTES INICIALES.....	36
TABLA 16: COSTES REALES	36
TABLA 17: ALGUNOS ERRORES INFORMÁTICOS DE LA HISTORIA.	77
TABLA 18: RESULTADOS DE LAS PRUEBAS EN SAMSUNG GALAXY S3	86
TABLA 19: RESULTADOS DE LAS PRUEBAS EN SAMSUNG GALAXY TAB 3.....	87
TABLA 20: FALLOS ENCONTRADOS	88

Capítulo 1

Introducción

1.1. Contexto y Motivación

Con la aparición de los Smartphone, las tecnologías móviles han experimentado un crecimiento exponencial, llegando a desbancar al ordenador personal como principal equipo informático para uso doméstico. Se ha entrado en la era Post-PC, en la que la informática se desvincula del ordenador como único soporte, pudiendo así expandirse a cientos de dispositivos diferentes.

Protagonista indispensable de este importante cambio conceptual que está afectando a la informática ha sido y es Android, comprado por Google en 2005, quien se ha encargado sabiamente de su posterior desarrollo. Este sistema operativo ha logrado facilitar la expansión de la informática, permitiendo que esta se integre en un teléfono, un televisor, un reloj, etc. Gracias a Android, la informática no tiene fronteras, estando presente en todos nuestros gadgets y dispositivos, naciendo así conceptos tan interesantes como el internet de las cosas, o las Smart Cities.

Ante este impresionante cambio tecnológico no deja al margen ningún ámbito. De este modo, la educación, que llevaba muchos años demandando una reforma en cuanto al modo de enseñar, se está beneficiando de las facilidades y posibilidades que estas tecnologías ofrecen. No en vano, se dice que estamos educando a niños del siglo XXI con profesores del siglo XX y métodos del siglo XIX. Sin embargo, hoy en día es posible sustituir las pesadas mochilas repletas de libros y cuadernos por una moderna Tablet, o complementar las densas charlas de los profesores con videos, audios, imágenes, diapositivas, etc. haciendo así la enseñanza mucho más fácil, divertida y amena.

En este marco nace Geolocalization Game, una aplicación diseñada para dispositivos Android, cuyo objetivo principal es ayudar y enseñar, de una manera didáctica, a localizar sitios en un mapa, mediante la creación de juegos de geolocalización. A diferencia de otras app, Geolocalization Game te permite crear tu propio juego personalizado, permitiendo al profesor hacer énfasis en aquellas localidades que quiera enseñar en cada momento.

1.2. Objetivos

El objetivo principal de este trabajo de fin de grado consiste en la creación de una aplicación móvil para Android que basado en Google Maps, genere juegos geográficos a medida. A partir de una lista de lugares que pueden localizarse con Google Maps, la aplicación generará un juego a medida para el apoyo al aprendizaje de dichos lugares geográficos.

Se generarán diferentes modos de juego que permitirán al usuario probar varios niveles de aprendizaje.

Teniendo en cuenta este objetivo principal, se proponen los siguientes objetivos secundarios:

- Estudio del entorno de desarrollo para Android, así como dicho lenguaje de programación.
- Diseñar y crear una interfaz que facilite el uso de la aplicación y que sea atractiva para los usuarios.
- Utilizar la API v3 de Google Maps para mostrar el mapa al usuario.
- Crear un sistema de varios niveles de juego para que el usuario obtenga un mejor aprendizaje.
- Permitir exportar un juego previamente creado.
- Desarrollar varios modos de juegos, para permitir que el usuario pueda mostrar sus conocimientos adquiridos de varias maneras.
- Permitir importar un juego.
- Al ser una aplicación destinada principalmente para el aprendizaje de niños es necesario que funcione sin datos personales de usuario.
- Fomentar el uso de aplicaciones informáticas en el ámbito educativo.
- Crear un sistema que permita al usuario crear su propio juego personalizado.

1.3. Resumen del contenido de la memoria

A continuación se describe brevemente las diferentes secciones por la que va estar compuesta la memoria del trabajo de fin de grado.

Introducción: Se describen los objetivos principales y secundarios del TFG además del entorno en que va a ser desarrollado y los principales factores que motivaron su desarrollo.

Entorno Tecnológico: En este capítulo se explica que tecnologías fueron necesarias comprender para la implementación del TFG.

Desarrollo de una aplicación en Android: Se explica el sistema operativo Android (características, historia, arquitectura, etc.). Se describe como está compuesta una aplicación en Android y sus principales características. Por último, se detalla la estructura de un proyecto de Android desarrollado con el SDK y Eclipse.

Gestión y planificación del proyecto: Se describe el posible alcance económico y real que abarca el proyecto, los posibles riesgos que pueden ir apareciendo así como la manera de mitigarlos. Además se muestra el calendario previsto y real que se ha llevado a cabo durante el proceso de desarrollo de TFG.

Requisitos: Se describen los diferentes tipos de requisitos y los casos de usos.

Análisis: Se describen los diferentes modelos de análisis que va a contener el TFG. Aquí podemos ver el modelo de dominio, los diagramas de secuencias relacionados con los casos de usos además de un diagrama de clases iniciales.

Diseño: Se explica la arquitectura global del sistema y los patrones usados en dicha arquitectura. Además se muestra el diseño detallado de los casos de usos anteriormente descritos.

Implementación: En este capítulo se describe cómo usar la API v3 de Google Maps desarrollada para JavaScript en Android. Además se describirá la manera de obtener los datos relacionados a una localidad a partir de su nombre usando los servicios de google.

Pruebas: Incluye las pruebas realizadas y los resultados obtenidos para la aplicación móvil desarrollada.

Conclusiones: Se explican las conclusiones obtenidas durante la realización del proyecto así como los objetivos que se han logrado cumplir. Además se mostraran las líneas futuras que pudieran mejorar la aplicación desarrollada.

Bibliografía: Se muestra la documentación que ha sido necesaria consultar en las distintas etapas del desarrollo de TFG.

Apéndices: Aquí podemos encontrar el Manual de Usuario, el glosario de términos además del contenido del CD.

Capítulo 2

Entorno Tecnológico

2.1 Tecnología para el uso de mapas en Android

En este apartado se presenta una breve descripción de las tecnologías necesarias para la implementación y desarrollo del TFG.

2.1.1. Necesidad del uso de mapas

El uso de cartografía en aplicaciones móviles se convertido en algo muy habitual en los últimos años. Gracias a esta tecnología, cualquier usuario puede trazar rutas para llegar a su destino, explorar un lugar al que tiene previsto viajar o utilizar la geolocalización para encontrar un teléfono robado o pedir ayuda en caso de emergencia.

2.1.2 Algunas Tecnologías disponibles

- **OpenStreetMap (OSM):** Es un proyecto colaborativo para la creación y edición de mapas libres que cuenta con mucha aceptación por parte de los usuarios. Según la Wikipedia en octubre de 2014 en el proyecto estaban registrados en torno a 1.840.000 usuarios de los cuales alrededor de 22.600 realizaban alguna edición en el último mes. Según estos datos podemos ver que OSM a pesar de ser software libre es un proyecto en el cual se trabaja activamente lo que favorece mucho su uso.
- **OpenLayers:** Es una biblioteca de java Script que al igual que OSM está implementada con código abierto. OpenLayers está orientado para navegadores web y ofrece un entorno interactivo. La comunidad de colaboradores que la compone es la encargada de su desarrollo y de su soporte.

- **Bing Maps:** Esta implementada por la compañía Microsoft y contiene un conjunto de herramientas que facilitan el trabajo con los mapas.
- **MapQuest:** Es un visualizador de mapas que además de poderse usar para la web tiene muy buena integración en dispositivos móviles.
- **Herramientas de ESRI:** Las aplicaciones y herramientas que se ponen a disposición son totalmente gratuitas, permite la inserción de los mapas tanto en web como aplicaciones móviles. Entre el conjunto de herramientas que tiene se encuentra ArcGIS Online que según su página web con una cuenta pública es posible crear, almacenar y administrar mapas, aplicaciones y datos, y compartirlos con los demás de forma gratuita.

2.2. Tecnologías utilizadas

2.2.1. Google Maps API v3:

Google Maps es un servidor de aplicaciones de mapas ofrecido por la empresa de Google. Según su página web la api de Google Maps está en más de 800 000 sitios y hay más de 250 millones de usuarios activos contando solamente los dispositivos móviles.

La API v3 es la última versión ofrecida por Google Maps para el uso de sus mapas por desarrolladores. Esta versión está implementada para ser usada mediante el lenguaje de programación Java Script, de aquí que su principal uso sea en integración de páginas web. No obstante Google ofrece una variedad de tutoriales para mostrarles a sus usuarios como se puede usar esta API en una aplicación Android. Además de la versión de pago, Google ofrece una licencia gratuita para el uso de su API, aunque limita el número de peticiones que se pueden hacer desde la app que la usa.

Google también ofrece una versión (API v2) para el uso de sus mapas en aplicaciones nativas de Android. La elección de usar la API v3 a pesar de no ser nativa para Android, se ha debido entre otras cosas a sus posibilidades de personalización de los mapas. De esta manera se ha conseguido mostrarle al usuario un mapa en el cual no se muestre el nombre de las localidades, ya que estas van tener que ser localizadas posteriormente. La versión v2 de la API de Google Maps no ofrecía esta facilidad, por lo cual su uso quedo totalmente descartado para el TFG.

2.2.2. HTML5:

Es la quinta versión del lenguaje de programación web HTML. Esta versión incorpora múltiples mejoras y una serie de elementos que reflejan el uso típico de los sitios web modernos.

En nuestro caso el uso de este lenguaje se hizo imprescindible para la realización de la app de Geolocalization Game ya que es la manera de poder usar el api v3 de Google Maps en una aplicación Android.

2.2.3.CSS

Es un lenguaje desarrollado para crear la presentación de las páginas web creadas con HTML o XML. Permite la creación de estilos que luego podrán ser asignados a distintas partes de la página web.

Mediante CSS se creó un estilo donde se configuro los márgenes y altura que contendría el objeto mapa. De esta manera ya tendríamos el tamaño del mapa dentro de la página web.

2.2.4.Java Script

Es un lenguaje de programación que aunque podemos encontrar su uso en documentos PDF o en aplicaciones de escritorio esta principalmente orientado a la programación web. Java Script permite mejoras en la interfaz de usuario así como un comportamiento dinámico de las páginas web. A pesar de que java Script y java pudieran confundirse ya que entre otras cosas comparten nombre, son lenguajes totalmente diferentes, por lo cual no comparten el mismo propósito ni tienen semánticas iguales.

En el TFG desarrollado a pesar de ser una app para Android, al usar la API v3 de Google Maps fue necesario el aprendizaje y uso de Java Script. Como se explicó anteriormente la API de v3 esta implementada para que su uso sea bajo Java Script. Una vez teniendo la página web que contendrá el mapa implementado con HTML5 y habiéndole dado el estilo al objeto que contendrá el mapa con CSS el nuevo paso es dar las funcionalidades a la página y de esto se encarga Java Script.

2.2.5.Java

Es un lenguaje de programación el cual fue creado con cinco objetivos principales en el cual podemos encontrar la programación orientada a objetos y la ejecución del mismo programa en múltiples sistemas operativos. Según la empresa consultora y de investigación de las tecnologías de la información Gartner en un artículo publicado en 2014 en su blog, Java era el lenguaje de programación más utilizado Aunque existen otras alternativas la gran mayoría de aplicaciones Android se desarrollan usando su lenguaje nativo ,Java. En nuestro caso hemos usado dicho lenguaje para crear una aplicación que fuera lo más nativa posible.

2.2.6.SQLite

Es un sistema de gestión de base de datos relacional de código libre que ofrece características muy interesantes como su pequeño tamaño, no requiere de un servidor además de ser fácil y de poca configuración. El conjunto de la base de datos puede ser guardado como un solo fichero en el dispositivo host, permitiendo almacenar hasta 2 Terabytes de tamaño en su versión 3. Android incorpora todas las funcionalidades necesarias de serie para la creación y gestión de base de datos SQLite.

Capítulo 3

Desarrollo de una aplicación en Android

3.1. El sistema operativo Android

En este capítulo se detallaran algunos aspectos importantes que son útiles conocer sobre el sistema operativo Android.

3.1.1. Introducción

Android es un sistema operativo de código abierto y gratuito para dispositivos móviles. Originalmente fue diseñado para dispositivos móviles con pantallas táctiles como las Tablet y teléfonos inteligentes (Smartphone), con el paso del tiempo Android ha conseguido implantarse en relojes inteligentes (Smartwatch), televisores, automóviles, etc. El sistema operativo Android está basado en un núcleo de Linux y los programas están escritos en Java, el cual es su lenguaje de programación nativo. Actualmente Android, gracias al éxito de los smartphones Samsung y a la gran compatibilidad que este presenta en terminales de otros fabricantes, lidera el mercado de los SO para dispositivos móviles.

3.1.2. Breve Historia

Android inicialmente fue desarrollado por la compañía de software Android Inc. La compañía fue adquirida por la multinacional Google en 2005 que junto a la Open Handset Alliance, fundada en 2007, llevan el desarrollo de Android desde entonces. En 2007 junto con la creación de la OHA se anuncia el sistema Android y google libera la mayoría del código de Android bajo la licencia Apache, una licencia libre y de código abierto. La primera versión comercial (de prueba) de Android se lanzada en septiembre de 2008.

Capítulo 3: Desarrollo de una aplicación en Android

El primer teléfono disponible en el mercado con el sistema operativo Android se dio a conocer al público el 22 de octubre de 2008 y era el HTC Dream. A principio de 2010 Google y HTC han colaborado junto para lanzar el Nexus One en dispositivos Android. Google ha continuado la comercialización de la gama Nexus y actualmente son estos dispositivos los que son utilizados para el desarrollo e implementación de Android y donde se estrenan las nuevas versiones disponibles.

Tanto el nombre de Android como el de Nexus One hacen alusión a la novela de Philip K. Dick ¿Sueñan los androides con ovejas eléctricas?, que posteriormente se llevaría al cine como Blade Runner.

El logo de Android es el robot Andy y aunque no está claro su origen se dice que su diseño estuvo basado en un video juego de los noventa llamado Gauntlet: The Third Encounter.

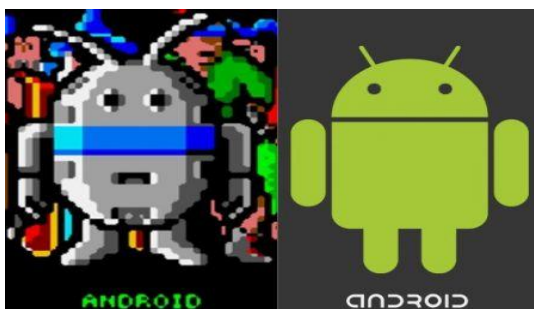


Figura 1: Origen y desarrollo
(Fuente [14])

3.1.3. Características y especificaciones actuales

La siguiente tabla muestra y detalla algunas de las características más importantes de Android.

Diseño de dispositivo	La plataforma es adaptable a pantallas de mayor resolución, VGA, gráficos 2D, gráficos 3D basada en OpenGL ES 2.
Conectividad	Entre otras, Android proporciona conectividad mediante: Bluetooth, Wi-Fi, LTE, HSDPA, HSPA+, NFC y WiMAX.GPRS,UMTS y HSDPA+.
Almacenamiento	Android utiliza como sistema de base de datos al liviano SQLite.

Navegador web	Basado en el motor de renderizado de código abierto WebKit y emparejado con el motor JavaScript V8 de Google Chrome.
Soporte de Java	Hasta la versión 4.4 Kitkat, Android utilizaba como máquina virtual de Java a Dalvik. Actualmente ha sido reemplazado por ART.
Soporte multimedia	Admite, entre otros, los siguientes formatos multimedia: 3GP, MP4, MPEG-4 , AMR, AAC, MP3,MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF y BMP.
Entorno de desarrollo	Cuenta con emulador de dispositivos, herramientas para depuración de memoria y análisis del rendimiento del software. El entorno oficial actualmente es Android Studio aunque también se suele usar el Eclipse con el plugging ADT.
Google Play	Es un repositorio de aplicaciones gratis y de pagos para sistemas Android.
Multi-táctil	Android tiene soporte nativo para pantallas capacitivas con soporte multi-táctil.
Multitarea	Multitarea real de aplicaciones está disponible, es decir, las aplicaciones que no estén ejecutándose en primer plano reciben ciclos de reloj.
Hardware adicional	Cámaras de fotos y vídeo, y sensores tales como GPS, acelerómetro, giroscopio, magnetómetro, sensores de proximidad y de presión, sensores de luz, termómetro, aceleración por GPU 2D y 3D.

Tabla 1: Principales características de Android

3.1.4. Nombre de las versiones y fragmentación

A continuación, se muestra un gráfico donde se detallan las versiones de Android a partir de la 2.2 (Froyo). También, se puede observar un gráfico de pastel con la distribución hasta febrero de 2015 de las distribuciones.

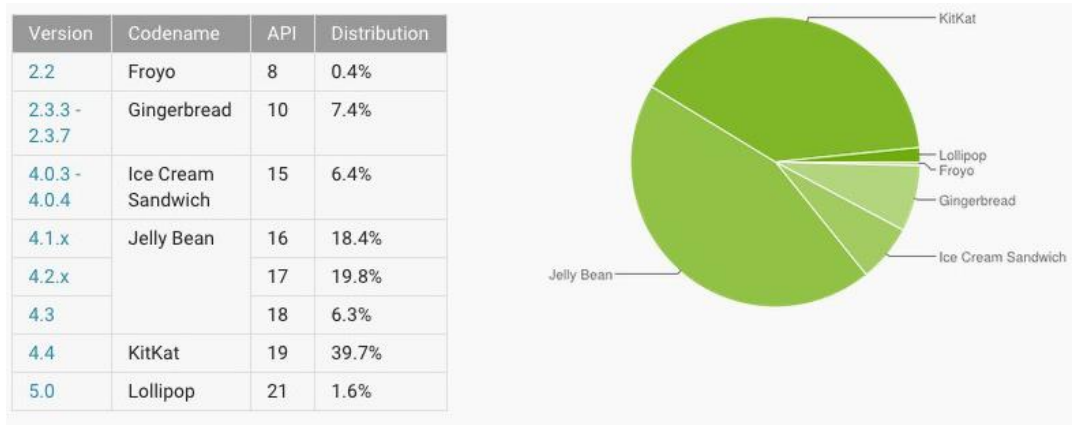


Figura 1: Recolección de datos en un periodo de 7 días hasta febrero del 2015

(Fuente [11])

La fragmentación de Android se debe principalmente a que las compañías de móviles, solo actualizan los dispositivos de gama alta.

Una de las ventajas más relevantes es que la fragmentación permite que Android este en múltiples dispositivos, tantos en dispositivos con características muy pobres hasta dispositivos de última generación. De esta manera Android puede estar presente tanto en un mercado para personas con pocos recursos económicos como en personas con muchos recursos. Estos factores junto a otros, hacen que Android sea el sistema operativo para dispositivos móviles más usado en los principales mercados del mundo y España.

Cuota de mercado de Android, iOS y Windows en España

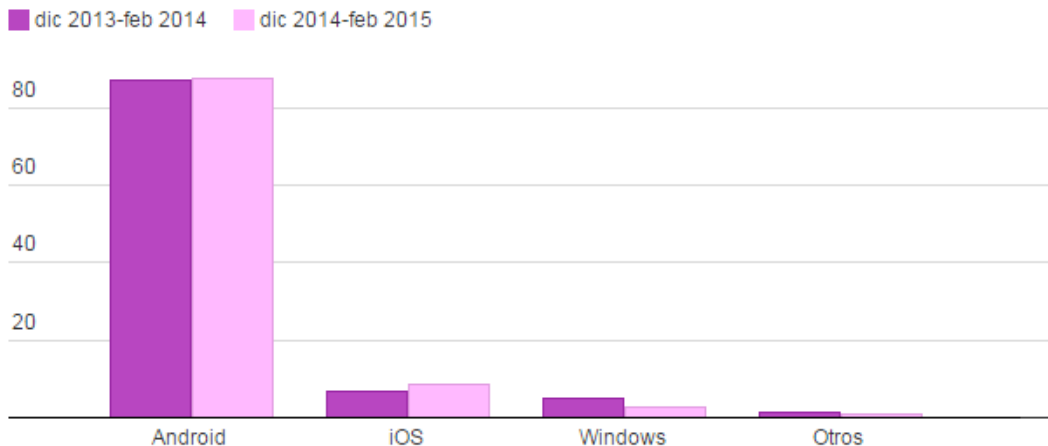


Figura 2: Recolección de datos entre diciembre de 2014 y febrero de 2015

(Fuente [12])

Cuota de mercado de Android, iOS y Windows en los principales mercados mundiales

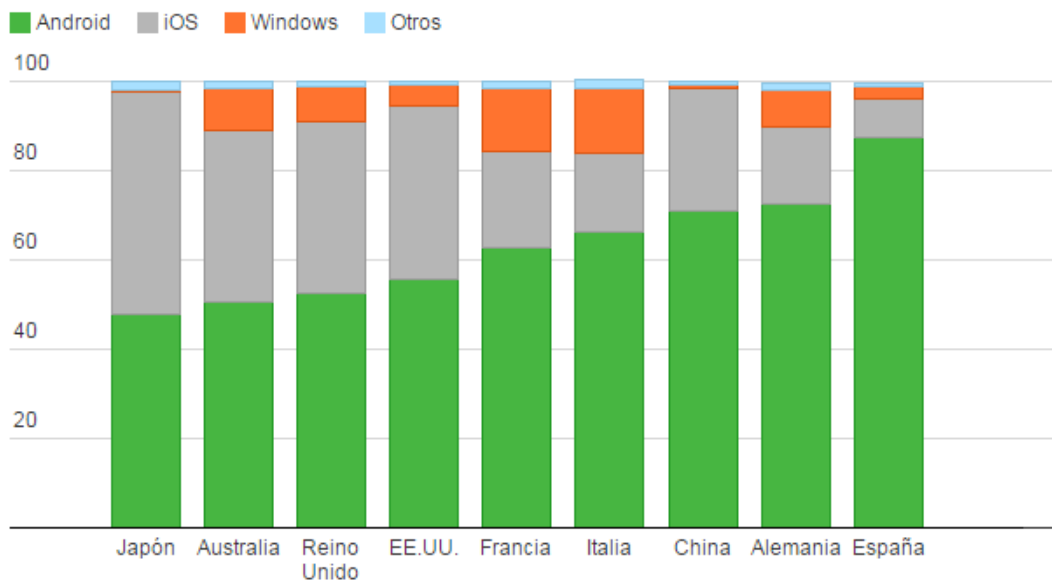


Figura 3: Recolección de datos entre diciembre de 2014 y febrero de 2015

(Fuente [12])

A pesar de la ventaja que la fragmentación presenta para que Android se posicione a la cabeza de los principales mercados del mundo, la fragmentación es un gran problema para Android, ya que esta afecta a muchos aspectos importantes como son a los desarrolladores de aplicaciones. Los programadores de app se encuentran con que tienen que adaptar sus programas para dispositivos con versiones de Android diferentes ya que estas no poseen las mismas características, y no solo esto sino que también tienen que tomar en cuenta los diferentes tamaños de pantallas al existir una gran variedad de ellas.

3.1.1.Arquitectura de Android

A continuación se muestra la Arquitectura de Android. Como se puede observar está formada por cuatro capas y una de las características más importantes que estas presentan es que todas están basadas en software libre.

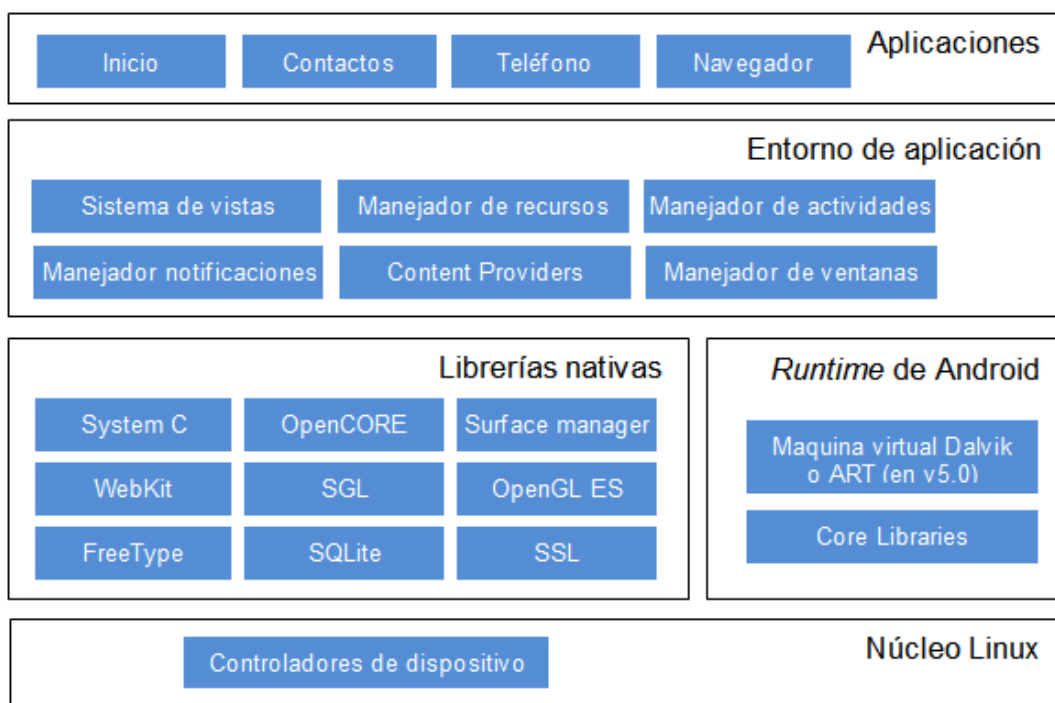


Figura 4: Arquitectura de Android

(Fuente [15])

- **Núcleo Linux:** Los servicios base de Android, como la gestión de memoria , seguridad y gestión de procesos dependen del sistema operativo Linux
- **Bibliotecas escritas en C:** Android cuenta con un conjunto de librerías que se exponen a los desarrolladores, a través de marcos de trabajos de aplicaciones. Estas bibliotecas son usadas también por varios componentes del sistema.

- **Runtime de Android:** Hasta la versión 4.4 Kitkat, Android utilizaba la máquina virtual Dalvik, que permite que cada aplicación corra su propio proceso. Dalvik, también ejecuta archivos en formato dex, el cual esta optimizado para memoria mínima. Actualmente, Dalvik ha sido reemplazado por ART que trae consigo una serie de mejoras como la recolección de basura, la mejora del rendimiento y las aplicaciones de depuración y perfilado. También, cuenta con un set de bibliotecas base que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje java.
- **Entorno de Aplicaciones (Framework Java):** La arquitectura está diseñada para que se puedan reutilizar componentes, por lo que cualquier app puede publicar sus capacidades y otras hacer uso de estas. Además, los desarrolladores tienen acceso completo a los mismos APIs de framwork usados por las aplicaciones base.
- **Aplicaciones:** Android viene con una serie de aplicaciones base como son el cliente de correo, mapa, gestor de ficheros entre otros. Las aplicaciones base están escritas en lenguaje Java.

3.2. Las Aplicaciones

3.2.1.Introducción

Las aplicaciones en Android se desarrollan habitualmente en Java con Android Software Development Kit (Android SDK). El SDK de Android tiene una serie de herramientas que permite compilar el código e incluir todos los recursos necesarios en un archivo comprimido con extensión apk (Application Package File), este formato es una variante del formato JAR de java que se usa para la distribución e instalación de las aplicaciones Android.

Una vez instalada la aplicación, cada una de ella tiene su propio sistema de seguridad de tal modo que cada app será un usuario diferente dentro de Android como sistema operativo basado en Linux multiusuario. Las app tendrá un ID de usuario Linux único y correrán su propio proceso, con su propia instancia de la máquina virtual Dalvik,

3.2.2.Herramientas para el desarrollo de Android

- **SDK** (Software Development Kit): Es la forma más habitual en la que se programan dispositivos Android, incluye un conjunto de herramientas de desarrollo como un depurador de código, una biblioteca, un simulador de dispositivos Android, documentación, ejemplo de código y tutoriales. Las actualizaciones del SDK están coordinadas con el desarrollo de Android aunque también es posible programar para versiones antiguas de Android.
- **NDK:** Permite la reutilización de código escrito en C/C++. Las aplicaciones se ejecuten directamente en el procesador, lo que hace que la ejecución de la aplicaciones sea en cierto modo más rápidas ya que no son interpretas por la máquina virtual. Las app que

suelen utilizar el NDK son aquellas que hacen un uso intensivo de la CPU, como por ejemplo los motores de videojuegos.

- **Phonegap:** Es un framework para el desarrollo de aplicaciones móviles de distintas plataformas teniendo como base un código genérico con herramientas tales como JavaScript, HTML, CSS. Las app que se generan no son nativa, sino híbridas ya que el renderizado se realiza mediante vistas web y no con interfaces gráficas específicas de cada sistema.
- **Xamarin:** Está basada en el lenguaje C# de Microsoft y en la plataforma NET. Permite generar la aplicación para Android con la extensión APK, por lo cual permita que corra de manera nativa. Proporciona acceso total a la API estándar de Android. Las aplicaciones en Xamarin ocupan más espacio que las nativas afectando al tiempo de descarga y a su almacenamiento.

3.2.3. Componentes

Existen cuatro diferentes tipos de componentes que podemos encontrar en una aplicación Android.

3.2.3.1. Las Activity

Según la página para desarrolladores de Android [], “una actividad (activity) es un componente de aplicación que proporciona una pantalla con la que los usuarios pueden interactuar con el fin de hacer algo, como marcar el teléfono, tomar una foto, envíe un correo electrónico, o ver un mapa. Cada actividad se da una ventana en la que dibujar su interfaz de usuario. La ventana normalmente llena la pantalla, pero puede ser más pequeña que la pantalla y flotar en la parte superior de otras ventanas.”

Generalmente una aplicación cuenta con varias actividades que están relacionadas entre sí. De esta manera una actividad puede iniciar otra con el objetivo de realizar acciones diferentes. Las actividades pueden pasar a segundo plano y cuando una activity en ejecución llama a otra, esta se detiene y se conserva en una pila ("back stack"). El sistema puede finalizar una actividad si esta no se encuentra en primer plano.

Una actividad puede existir esencialmente en tres estados:

- **Resumed:** La actividad está encima de la pila, por lo cual es visible y está en primer plano de la pantalla (Este estado también se refiere a veces como "Running").
- **Paused:** La actividad es visible pero no tiene el foco. Esto pasa cuando otra actividad pasa a primer plano pero no ocupa toda la pantalla o es parcialmente transparente. Una actividad en pausa se conserva en memoria y mantiene toda su información pero puede ser eliminada por el sistema en situaciones donde la memoria sea extremadamente baja.
- **Stopped:** La actividad en este estado ya no es visible para el usuario y el sistema pudiera eliminarla si se necesitara memoria en otro lugar.

El paso de estado de una actividad va a generar eventos que pueden ser capturados por ciertos métodos de las actividades:

- **onCreate(Bundle):** Se está creando la actividad. Se utiliza para crear todo tipo de inicializaciones, como la creación de la interfaz de usuario.
- **onStart():** La actividad está a punto de hacerse visible para el usuario.
- **onResume():** La actividad se ha hecho visible. El código en este método deber ser ligero para evitar transiciones lentas que hagan esperar al usuario.
- **onPause():** Otra de las actividades está tomando el foco y que la actividad actual va a pasar a segundo plano. Este método es un buen lugar para salvar los datos que están en edición o detener animaciones y música. El código en este método deber ser ligero para evitar transiciones lentas que hagan esperar al usuario.
- **onStop():** La actividad ya no es visible.
- **onRestart():** La actividad va a volver a ser lanzada habiendo pasado por onStop().
- **onDestroy():** La actividad va a ser totalmente destruida.

A continuación, se muestra el ciclo de vida de una actividad:

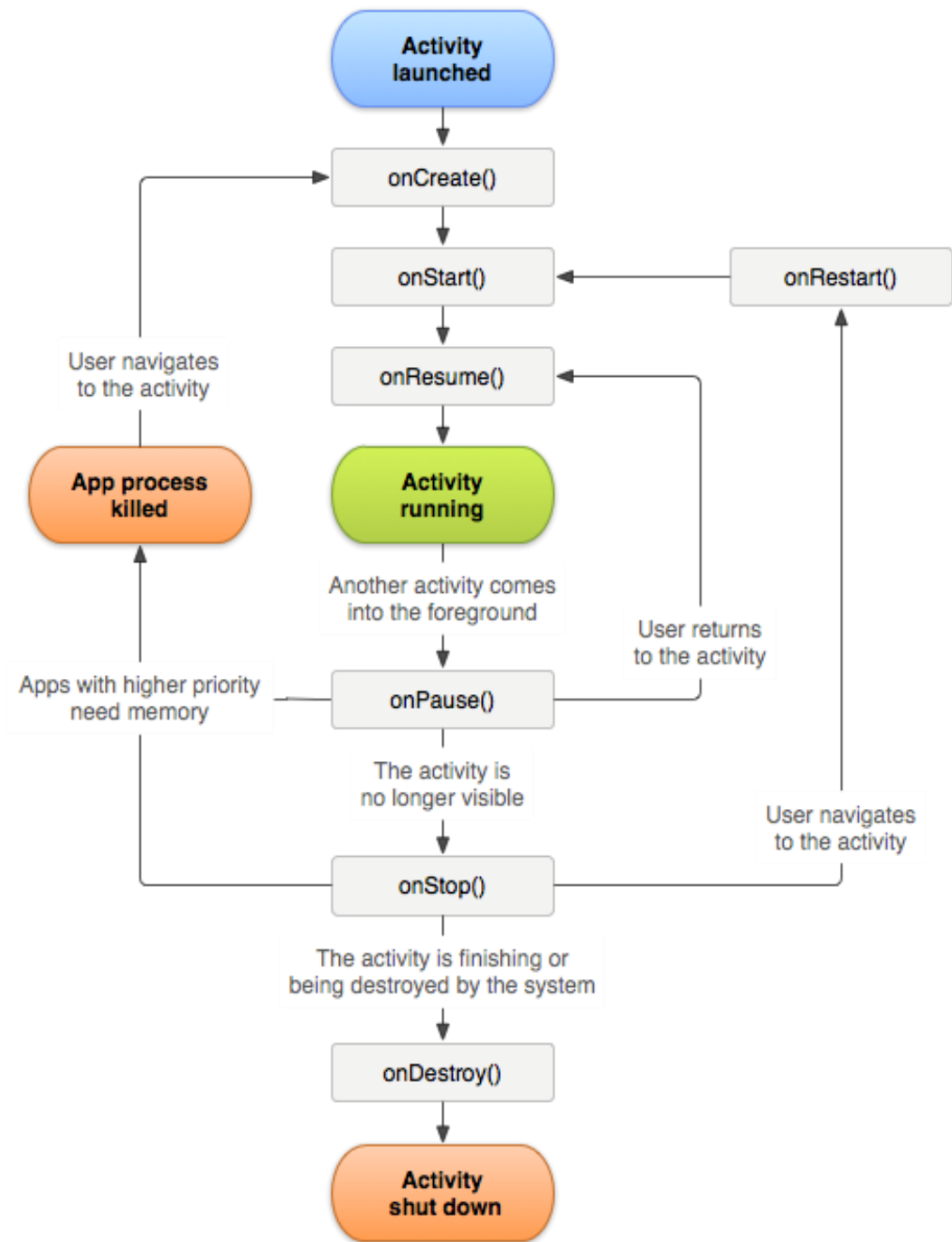


Figura 5: El ciclo de vida de una actividad

(Fuente [2])

3.2.3.2. Los Service

Un servicio es un componente de aplicación que se ejecuta en segundo plano sin que el usuario tenga que interactuar con este. Son usados principalmente para tareas de larga duración. Los servicios pueden seguir ejecutándose incluso si el usuario cambia de aplicación. Un ejemplo de uso de los servicios es un reproductor de música. La reproducción se iniciara desde una actividad y la música se debe seguir escuchando cuando el usuario quite el foco de la aplicación y este usando otras distintas.

3.2.3.3. ContentProvider

Este componente es el mecanismo que implementa Android para compartir información entre aplicaciones. Se le pueden asociar una serie de permisos, de esta manera se puede limitar las operaciones que otras aplicaciones realizan sobre estos. El sistema operativo Android incorpora una serie de aplicaciones con ContentProvider como son los contactos, calendario y mensajes. Con algunas restricciones estos proveedores son accesibles para cualquier aplicación.

3.2.3.4. BroadcastReceiver

Nos permite reaccionar a eventos globales del sistema. De esta manera podemos configurar nuestra aplicación para que reaccione a diferentes eventos, como puede ser la batería baja, el apagado de la pantalla o el cambio de localización de esta manera. El receptor tiene que estar suscrito al emisor del evento. Aunque no tienen relación directa con el usuario, pueden crear notificaciones que serán visibles en el dispositivo. Los widget son ejemplos de receptores de eventos.

Intent:

Para poder invocar componentes, se usa un mecanismo llamado Intent. Los Intent se pueden entender como los mensajes o peticiones que son enviados entre componentes de una misma aplicación o entre distintas aplicaciones. En concreto, se usan cada vez que queremos lanzar una actividad (`startActivity()` y `startActivityForResult()`), un servicio (`startService()`) o los anuncios de tipo broadcast (`sendBroadcast()`). Cuando se crea un objeto Intent, este contiene información de interés para que el sistema trate adecuadamente la intención (Intent) o para el componente que recibe la intención (Intent)

Los Intent pueden incluir:

- **nombre** del componente: De esta manera podemos identificar el componente que queremos lanzar.
- **Acción:** Una cadena de caracteres donde indicamos la acción a realizar. Si es un receptor indicaríamos la acción que tuvo lugar y queremos reportar. Para acciones genéricas la clase Intent define una serie de constantes como `ACTION_CALL` que inicializa una llamada de teléfono o `ACTION_BATTERY_LOW` que advierte que la batería esta baja, este último se utiliza para receptores de anuncios.
- **Categorías:** Es un complemento de la acción e indica información adicional sobre el tipo de componente que ha de ser lanzado. Al igual que la acción, la clase Intent define una serie de categorías genéricas que se pueden utilizar como, por ejemplo, `CATEGORY_PREFERENCE`, que significa que la actividad a lanzar es un panel de preferencias o una muy usada como `CATEGORY_LAUNCHER`, que significa que la

actividad puede ser la actividad inicial de una tarea y se muestra en el lanzador de aplicaciones de nivel superior.

- **Datos:** Mediante una URI se hace referencia a los datos con los que trabajaremos. Para indicar el tipo de dato que estamos trabajando se usa el tipo MIME, que va asociado a la URI .Ejemplos de tipos MIME son text/xml, audio/mp3, imagen/jpeg,
- **Extras:** Mediante un objeto de la clase Bundle se puede enviar información extra que será recibida por el componente que vamos a lanzar.

Los Intent pueden ser de dos tipos:

Explícitos: Llevan indicado explícitamente el componente a lanzar. Se utilizan para lanzar componentes internos de la aplicación.

Implícitos: No especifican un componente en especial, sino que le pregunta al sistema que servicio o componente es el más adecuado para realizar la petición. Por ejemplo, se pueden utilizar para abrir la cámara o enviar un SMS.

3.3. Estructura de un proyecto en Android

3.3.1.Introducción

Si creamos un proyecto Android usando las herramientas del SDK estos tendrán una estructura específica. De esta manera se garantiza una compilación y un empaquetado correcto del proyecto.

Existen tres tipos diferentes de proyectos Android, que tienen la misma estructura pero se usan para funciones diferentes.

Proyectos Android: Es el principal tipo de proyecto y contiene el código fuente de la aplicación, archivos de recursos, etc. Todo esto será compilado por el SDK y generara un archivo apk que finalmente es el que se instalará en el dispositivo.

Proyectos de prueba: Es un proyecto que estará vinculado al proyecto que contiene la aplicación que queremos probar, estos proyectos contienen el código necesario para poder realizar las pruebas.

Proyectos de bibliotecas: Son proyectos que se utilizan habitualmente cuando se quiere reutilizar un código. No son instalables, sino que contiene recursos y código fuente que será utilizado por otros proyectos.

3.3.2.Estructura

Cuando iniciamos la creación de un nuevo proyecto Android en eclipse usando el SDK se inicializa el asistente de creación del proyecto que nos guiara por las distintas opciones de creación y configuración.

Entre las opciones que podemos configurar desde el asistente se encuentra el nombre de la aplicación, la versión de SDK mínima, el tipo de proyecto y el icono para la aplicación. Una vez

finalizado el asistente se generara el proyecto Android que contendrá los siguientes archivos y directorios:

- **src/:** Contiene las clases java, las actividades (activity) y todos los archivos de código fuente. Todo esto puede estar distribuidos en paquetes.
- **gen/:** Contiene los archivos generados por las herramientas de desarrollo. Aquí podemos encontrar el fichero R.java que contiene las referencias a los recursos utilizados por la aplicación como pueden ser los componentes de una vista.
- **assets/:** Inicialmente este directorio está vacío .Se puede utilizar para cargar ficheros en tiempo de ejecución como una base de datos inicial o un fichero .html. Los archivos incluidos en este directorio no se les genera una referencia en el fichero R.java.
- **bin/:** En este directorio se encuentran los archivos compilados de la aplicación otros ficheros auxiliares .Además de esto se encuentra el fichero apk.
- **libs/:** Se pueden almacenar bibliotecas externas privadas que utilicemos en nuestra aplicación.
- **res/:**Contiene recursos de la aplicación:

anim/: Contiene los ficheros XML con las animaciones utilizadas por la aplicación.

color/: Podemos guardar ficheros XML con las definiciones de los colores.

drawable/: Generalmente se usa para guardar las imágenes de la aplicación en diferentes resoluciones. Aquí podemos encontrar el icono de la aplicación.

layout/: Contienen los ficheros XML con las definiciones de las interfaces de usuario. Aquí se almacena las vistas relacionadas con una actividad. También se puede crear otra carpeta con el nombre layout-land para que el sistema automáticamente muestre otra interfaz de usuario si el dispositivo esta en horizontal.

menú/: Contiene las definiciones de los menús de la aplicación en formato XML.

raw/: En este directorio se puede guardar recursos adicionales como son la música, videos o archivos de textos. Estos recursos pueden ser referenciados mediante la clase R.java.

values/: Permite guardar otros recursos de la aplicación que pueden ser referenciados por la clase R.java, como cadenas de texto (strings.xml) o colores (colors.xml). Son muy útiles para traducir una aplicación a varios idiomas.

xml/: Contiene otros ficheros de datos requeridos por la aplicación.

- **AndroidManifest.xml:** Este es probablemente uno de los ficheros más importantes de la aplicación. Aquí se describe la configuración de la aplicación, se definen las intenciones, servicios y proveedores de contenido. Aquí también se definen los permisos que serán solicitados al usuario hacia recursos del dispositivo como puede ser el acceso a los datos o a internet.
- **project.properties:** Este fichero se genera automáticamente y sirve entre otras cosas para comprobar la versión de la API cuando la aplicación es instalada en el dispositivo.

- **ic_launcher-web.png**: Es el icono de la aplicación en una resolución de 512x512 para ser usado en páginas web.
- **proguard-project.txt**: Permite optimizar y ofuscar el código generado.



Figura 6: Estructura de un proyecto Android en Eclipse

(Fuente [15])

3.3.3. El fichero AndroidManifest.xml

Se encuentra en la raíz de la aplicación y contiene la configuración de la aplicación. Toda aplicación debe tener este fichero. A pesar de que podemos realizar la configuración de este fichero

mediante una interfaz gráfica es recomendable hacerlo desde el propio XML. A continuación vamos a nombrar y describir algunos aspectos importantes de este fichero.

- Cada vez que realizamos una publicación de la aplicación debemos ir cambiando el número de versión. Esto se realiza desde el AndroidManifest.xml.
- Se define el icono y nombre de la aplicación, este último debe estar en el fichero strings.xml de la carpeta "values".
- Todas las actividades tienen que ser declaradas aquí.
- Se definen los permisos que va a solicitar la aplicación para su correcto funcionamiento.
- Se definen las pantallas soportadas y compatibles.
- Se declaran la configuración de las versiones soportadas.

La siguiente imagen muestra un fichero AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="net.ivanvega.Archivos"
    android:versionCode="1"
    android:versionName="1.0">
    <uses-sdk android:minSdkVersion="10" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".MainActivity"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Figura 7: Fichero AndroidManifest.xml

Capítulo 4

Gestión y planificación del proyecto

4.1. Introducción

Para el desarrollo de este proyecto hemos utilizado la metodología del Proceso Unificado. Las principales características de esta metodología es que se caracteriza por estar dirigida por los casos de usos, está centrada en la arquitectura y es un proceso iterativo e incremental.

El Proceso Unificado puede ser dividido en cuatro fases:

- **Inicio:** Se define el negocio. Se definen aspectos importantes como los costes del proyecto, los plazos, la viabilidad y se presenta un modelo.
- **Elaboración:** Se obtiene la visión refinada del proyecto. Se realizan tareas como la implementación iterativa del núcleo, la obtención de nuevos requisitos y el ajuste de las estimaciones.
- **Construcción:** Se construye la totalidad del sistema, incluyendo los requisitos mínimos.
- **Transición:** En esta fase la aplicación debe de estar lista para ser usada por el usuario final sin ningún problema. Una vez finalizada esta fase , se debe de empezar a pensar en mejoras de la aplicación

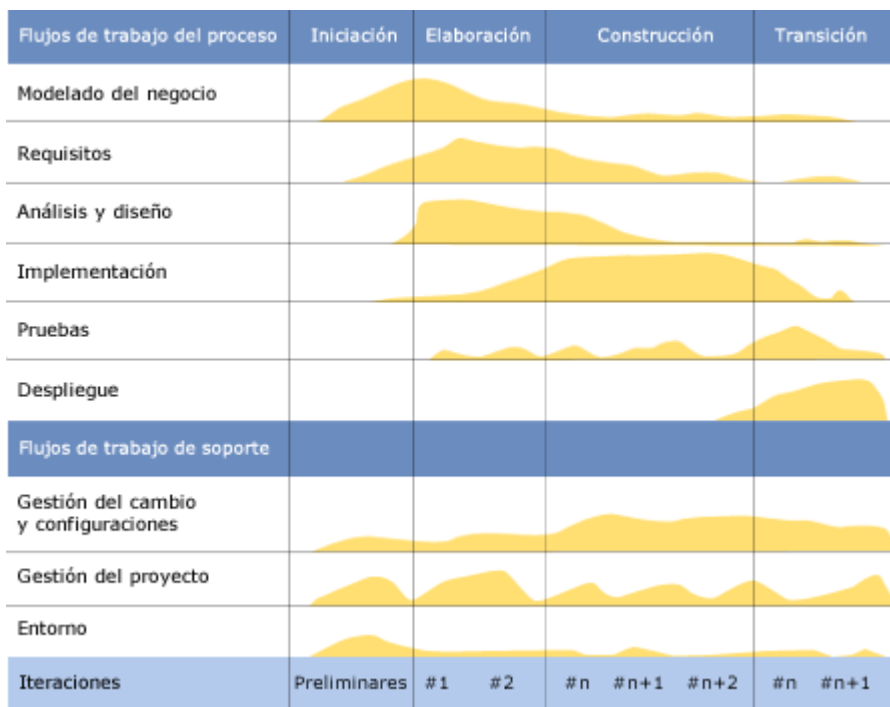


Figura 8: Proceso Unificado

(Fuente [17])

4.2. Gestión de riesgos

"La negación es una táctica común que sustituye a la ignorancia deliberada de la planificación cuidadosa" (Charles Tremper).

Según el Project Management Institute, la gestión de proyectos (o project management) se define como la aplicación del conocimiento, habilidades, herramientas y técnicas a las actividades de un proyecto, a fin de cumplir los requerimientos de este.

El proceso llevado a cabo por la gestión de riesgos contiene los procesos relacionados con la identificación, análisis, y planes de contingencia ante los posibles riesgos. Todo esto se puede dividir en las siguientes fases:

- **Identificación:** Posibles riesgos que puedan surgir en las diferentes etapas de la elaboración del proyecto.
- **Análisis de los riesgos:** Debe de exponerse la probabilidad de que estos ocurran y la repercusión que estos podrían tener.
- **Planes de contingencia:** Contiene la manera a actuar en caso de que se dieran los riesgos. El objetivo principal de este plan de contingencia es evitar los riesgos o mitigarlos en el peor caso.

- **Supervisión y control de riesgos:** Se realiza un seguimiento de los riesgos que se han dado y las consecuencias que este ha traído para el desarrollo del proyecto.

4.2.1. Identificación y análisis de riesgo

Para un mejor análisis de los riesgos se han usado los siguientes criterios:

Tipos de riesgos:

- Riesgos de proyecto: Amenazan al plan de proyecto. Si este riesgo apareciera pudiera aumentar los costes o el tiempo de desarrollo del proyecto.
- Riesgos técnicos: Están relacionados con los aspectos técnicos como pueden ser la implementación y el diseño de la aplicación. Este tipo de riesgos amenaza a la calidad y a la planificación.
- Riesgos de producto: Amenazan a la viabilidad del sistema a construir.

Probabilidad de los riesgos:

- Muy alta (>75%)
- Alta (50-75%)
- Media (30-50%)
- Baja (10-30%)

Consecuencia de los riesgos en orden:

- Catastrófico
- Critico
- Tolerable
- Insignificante

A continuación se muestran los riesgos identificados:

Riesgo	Tipo de riesgo	Probabilidad de ocurrencia	consecuencias
R01- Se necesita más tiempo para el estudio de la tecnología a usar.	Proyecto	Muy alta	Crítico
R02- Falta de motivación.	Proyecto	Baja	Tolerable
R03- Examen extraordinario de física.	Proyecto	Alta	Crítico
R04- Realización de las prácticas de empresa.	Proyecto	Muy alta	Tolerable
R05- Cambios de los requisitos funcionales una vez empezado el proyecto.	Proyecto	Media	Crítico
R06-Planificación muy optimista.	Proyecto	Alta	Crítico
R07- Fallos al probar la aplicación en dispositivos con características diferentes.	Técnicos	Alta	Crítico
R08- Desarrollo de Interfaces incorrectas.	Producto	Alta	Crítico

Tabla 2: Identificación de riesgos

4.2.2. Plan de riesgos

A continuación se muestran las siguientes tablas, donde para cada uno de los riesgos identificados anteriormente se elabora un plan con el objetivo de evitarlos o mitigarlos.

R01- Se necesita más tiempo para el estudio de la tecnología a usar.	
Descripción	Puede que se trate de usar tecnologías que no se ajusten a las necesidades del proyecto y haya que cambiar a otras que no se dominen.
Fase/actividad	Se puede dar en cualquier momento del desarrollo.
Plan de Mitigación	Se ajustará la planificación con el objetivo de dedicarle más tiempo al estudio de la tecnología.
Plan de Contingencia	Si el tiempo de estudio de la tecnología pasa de 30 días se retrasara la entrega del proyecto a septiembre.

Tabla 3: R01- Se necesita más tiempo para el estudio de la tecnología a usar.

R02- Falta de motivación.	
Descripción	La falta de motivación puede provocar que el rendimiento durante las horas dedicada sea bajo, esto provocaría un retraso en el desarrollo de proyecto. Puede producirse a causa de algún requisito que no se consiga o a un horario no ajustado.
Fase/actividad	Se puede dar en cualquier momento del proyecto.
Plan de Mitigación	Se ajustara el calendario para que se trabaje a una hora donde se consiga mejor motivación.
Plan de Contingencia	Si no se consigue un mejor aprovechamiento de las horas a causa de no conseguirse un requisito, se tratara de mejorar algunos aspectos del proyecto para conseguir una mejor motivación.

Tabla 4: R02- Falta de motivación.

R03- Examen extraordinario de física.	
Descripción	Puede ser necesario ajustar el calendario para dedicarle tiempo a estudiar Física.
Fase/actividad	Se puede dar a partir de la etapa de implementación.
Plan de Mitigación	Se ajustara el calendario del proyecto para poder dedicarle tiempo a la asignatura de Física.
Plan de Contingencia	Si no se consigue aprobar Física se retrasara el proyecto hasta julio del siguiente año.

Tabla 5: R03- Examen extraordinario de física.

R04- Realización de las prácticas de empresa.	
Descripción	Debido a la realización de las prácticas de empresas puede que el calendario dedicado al inicio del proyecto no sea el mismo.
Fase/actividad	Se puede durante la etapa de implementación o antes.
Plan de Mitigación	Se ajustara el calendario del proyecto según el horario de las prácticas de empresas.
Plan de Contingencia	Si las prácticas de empresa fueran en horario de mañana y de tarde y terminaran a principio de junio entonces se alargaría el proyecto hasta septiembre.

Tabla 6: R04- Realización de las prácticas de empresa.

R05- Cambios de los requisitos funcionales una vez empezado el proyecto.	
Descripción	No se puede realizar algún requisito funcional debido a que no se ajuste la tecnología usada.
Fase/actividad	Durante la etapa de implementación.
Plan de Mitigación	Se ajustaran los requisitos de modo que se puedan utilizar con la tecnología usada.
Plan de Contingencia	Se modificara dicho requisito de modo que sea posible su implementación.

Tabla 7: R05- Cambios de los requisitos funcionales una vez empezado el proyecto.

R06-Planificación muy optimista.	
Descripción	Debido a la inexperiencia en la planificación de proyectos puede que se necesite más tiempo que el planificado en algunas etapas del proyecto.
Fase/actividad	Se puede dar en cualquier etapa del proyecto.
Plan de Mitigación	Se ajustara el calendario para dedicarle más tiempo a las posibles etapas mal planificadas.
Plan de Contingencia	Se alargara la entrega del proyecto.

Tabla 8: R06-Planificación muy optimista.

R07- Fallos al probar la aplicación en dispositivos con características diferentes.	
Descripción	Puede que una vez desarrollada la aplicación, cambie el aspecto y el funcionamiento de esta en dispositivos Android con características diferentes como la pantalla.
Fase/actividad	Etapa de despliegue de la aplicación.
Plan de Mitigación	Se diseñara y se probara la aplicación, con diferentes dispositivos durante la implementación de la aplicación.
Plan de Contingencia	En caso de que no se pueda adaptar a ciertos dispositivos, estos serán puestos como dispositivos no compatibles en la versión de la app desarrollada.

Tabla 9: R07- Fallos al probar la aplicación en dispositivos con características diferentes.

R08- Desarrollo de Interfaces incorrectas.	
Descripción	Debido al mal diseño de la interfaz, puede que exista una mala comunicación entre estas o que la experiencia del usuario no sea buena.
Fase/actividad	Puede ocurrir durante la etapa de Diseño, Implementación o durante las pruebas de la aplicación
Plan de Mitigación	Se realizaran pruebas de interfaz con posibles usuarios y se dedicara más tiempo a este aspecto.
Plan de Contingencia	Se rediseñara la interfaz, siempre tratando de mantener el mayor número posible de funcionalidades ya implementadas. El objetivo fundamental es solucionar el problema y aprovechar el código ya desarrollado.

Tabla 10: R08- Desarrollo de Interfaces incorrectas.

4.2.3. Supervisión y seguimiento de riesgos

Durante la etapa de desarrollo del TFG se ha llevado un seguimiento y gestión de los riesgos que han ido apareciendo. A continuación, se detallan los riesgos que se han producido finalmente.

Riesgo	Detalle
R01- Se necesita más tiempo para el estudio de la tecnología a usar.	La Api elegida para el tratamiento del mapa carecía de funcionalidades requeridas. El cambio de Api produjo retraso principalmente en el estudio de la tecnología. Fue necesario realizar una programación hibrida y se carecía de los conocimientos para realizar la comunicación entre los diferentes lenguajes. Fue necesario aumentar el tiempo dedicado para adquirir dichos conocimientos.
R03- Examen extraordinario de física.	Fue necesario el examen de Física, lo que produjo que se retrasara considerablemente la fase de implementación. Debido a esto fue necesario retrasar la entrega del proyecto hasta septiembre.
R04- Realización de las prácticas de empresa.	Durante las prácticas de empresas el tiempo dedicado al proyecto se vio afectado y se aumentó la cantidad de días dedicados a la fase de implementación.
R06-Planificación muy optimista.	Mala planificación durante la etapa de implementación. Fue necesario aplazar la entrega del proyecto
R07- Fallos al probar la aplicación en dispositivos con características diferentes.	Problemas en el uso del tutorial de la app en Tablet de 7 y 10 pulgadas. Este riesgo ocasionó un retraso en la implementación del proyecto.

Tabla 11: Supervisión y seguimiento de riesgos.

4.3. Planificación

4.3.1. Recursos

Recursos humanos:

- Este proyecto será desarrollado por Maykel García Olivares como proyecto de fin de carrera.
- La Dra. Yania Crespo González será la encargada de supervisar y controlar el desarrollo del proyecto.

Recursos Técnicos:

- Hardware:
 - Ordenador portátil Lenovo G500s
 - Dispositivo móvil Samsung Galaxy S3
 - Tablet Samsung Galaxy Tab 3 de 7 pulgadas

- Software:
 - Eclipse IDE for Java Developers Version: Kepler Service Release 2
 - Astah Professional 6.9.0
 - Microsoft Project Professional 2013 SP 1
 - SDK y ADT para eclipse con todos los plugin necesarios para desarrollar la app.
 - Pivotal Tracker
 - Plugin de Google Maps JavaScript API v3.
 - Sistema Operativo Windows 7 Professional Service Pack 1 64 bits
 - Microsoft Word 2013

4.3.2. Estimación temporal

Enumeración de tareas:

- Propósito General ,alcance y objetivos del proyecto
- Obtención y análisis de los requisitos funcionales y no funcionales
- Planificación: Se realizara la estimación del tiempo y recursos necesarios para la elaboración del proyecto. Una vez con estos datos se deberá realizar un presupuesto económico para dicho desarrollo.
- Estudios Iniciales: Se realizara un estudio sobre las arquitecturas a usar. Especialmente se elegirá una API para el uso de mapas que cumpla con los requisitos necesarios para poder ser usada durante la implementación.
- Fase de Análisis.
- Fase de Diseño.
- Implementación.
- Pruebas
- Revisión de las tareas antes realizadas y corrección de posibles problemas.
- Generar la documentación correspondiente al usuario (Manual de Usuario).
- Realizar la memoria del proyecto

Al finalizar cada fase se deberá entregar un documento con dicho contenido.

4.3.3. Distribución temporal

Previsiones Iniciales:

Para estas previsiones se cuenta con la realización de las Prácticas de Empresas totalmente ajenas al Proyecto de Fin de Carrera. Además, se cuenta como fecha inicial de proyecto el día 10 de febrero de 2015 que cuando inicia el segundo cuatrimestre escolar y como fecha de finalización se obtiene el día 1 de julio de 2015 ya que dicho día es la presentación de la solicitud de defensa.

El calendario de horas de trabajo equivalentes por días es el siguiente:

- Lunes a Viernes: de 16:00 hasta las 22:00
- Sábado: de 9:00 a 13:00

Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
Propósito General ,alcance y objetivos del proyecto	4 días	mar 10/02/15	sáb 14/02/15	
Obtención y análisis de los requisitos funcionales y no funcionales	5 días	sáb 14/02/15	sáb 21/02/15	1
Planificación	3 días	sáb 21/02/15	mié 25/02/15	2
Estudios Iniciales	4 días	mié 25/02/15	mar 03/03/15	3
Análisis	8 días	mar 03/03/15	sáb 14/03/15	4
Diseño	14 días	sáb 14/03/15	vie 03/04/15	5
Implementación	45 días	sáb 04/04/15	sáb 06/06/15	6
Pruebas	3 días	lun 08/06/15	jue 11/06/15	7
Revisión Y Corrección	5 días	vie 12/06/15	vie 19/06/15	8
Manual de Usuario	3 días	vie 19/06/15	mar 23/06/15	9
Realizar la memoria del proyecto	5 días	mar 23/06/15	mar 30/06/15	10

Tabla 12: Duración estimada de las tareas

Capítulo 4: Gestión y planificación del proyecto

A continuación se muestra el diagrama de Gantt relacionado con la tabla anterior:

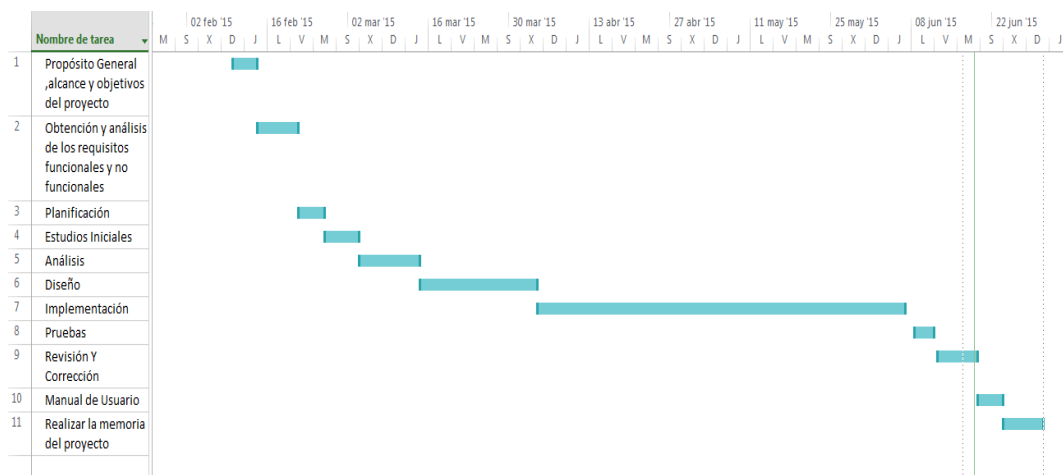


Figura 9: Diagrama de Gantt de la calendarización estimada

Desviación:

Debido a los riesgos producidos, la planificación del TFG se ha visto afectada y se ha tenido que retrasar la fecha de entrega. A continuación, se muestra la calendarización real del proyecto, la desviación de esta sobre su previsión y las causas que la han originado.

Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
Propósito General, alcance y objetivos del proyecto	4 días	mar 10/02/15	sáb 14/02/15	
Obtención y análisis de los requisitos funcionales y no funcionales	5 días	sáb 14/02/15	sáb 21/02/15	1
Planificación	6 días	sáb 21/02/15	lun 02/03/15	2
Estudios Iniciales	10 días	lun 02/03/15	lun 16/03/15	3
Análisis	12 días	lun 16/03/15	jue 02/04/15	4
Diseño	17 días	jue 02/04/15	lun 27/04/15	5
Implementación	63 días	lun 27/04/15	sáb 25/07/15	6
Pruebas	5 días	sáb 25/07/15	sáb 01/08/15	7
Revisión Y Corrección	6 días	lun 03/08/15	mar 11/08/15	8
Manual de Usuario	5 días	mar 11/08/15	mar 18/08/15	9
Realizar la memoria del proyecto	8 días	mar 18/08/15	sáb 29/08/15	10

Tabla 13: Duración real de las tareas

Nombre de tarea	Duración	Días Incrementados	Principales Causa de la desviación
Propósito General ,alcance y objetivos del proyecto	4 días	0	
Obtención y análisis de los requisitos funcionales y no funcionales	5 días	0	
Planificación	6 días	3	Análisis sobre la duración de las tareas
Estudios Iniciales	10 días	6	Búsqueda de un mapa que se adecuara a las necesidades
Análisis	12 días	4	Redistribución de las clases a usar.
Diseño	17 días	3	Realización de correcciones y uso de plataforma Android
Implementación	63 días	18	Dificultades con la programación hibrida debido al uso del mapa de Java Script
Pruebas	5 días	2	Necesidad de aumentar el número de usuarios y de dispositivos donde se realizaron las pruebas
Revisión Y Corrección	6 días	1	Corrección de errores
Manual de Usuario	5 días	2	Corrección de errores
Realizar la memoria del proyecto	8 días	3	Corrección de errores

Tabla 14: Horas incrementadas de las tareas

La idea de usar el Api v3 de Google Maps con una interfaz totalmente de Android hizo q la cantidad de días de trabajo aumentaran considerablemente ante lo previsto.

4.3.4. Costes

A continuación, se detallan los costes previstos teniendo en cuenta tanto los recursos técnicos como humanos. En el caso de los recursos humanos se ha usado el “Convenio Colectivo de Oficinas y Despachos Valladolid 2015-2018” para un grupo profesional IV de nivel II para obtener la cantidad de euros/horas que gana el trabajador. Además para disminuir el coste total del proyecto se ha optado por usar software de licencia gratuita u ofrecida por la Universidad de Valladolid. Debido que el coste de este tipo de software es 0 no se ha incluido en la tabla de costes iniciales.

Costes iniciales:

Se ha optado por hacer un ajuste aplicado como prevención a los posibles riesgos que pudieran aparecer debido a la inexperiencia tanto a la hora de planificar como a un posible incremento en las horas del trabajo a causa de la tecnología usada durante la implementación.

Concepto	Descripción	Importe en Euros
Costes del trabajador	99 días x 6 h/día * 11,5 euros/día	6831
Equipamiento	Licencia de Microsoft Office 2013	135
	Ordenador portátil Lenovo G500s	799
	Dispositivo móvil Samsung Galaxy S3	219
	Tablet Samsung Galaxy Tab 3 de 7 pulgadas	120
Coste total inicial		8104
Ajuste aplicado	25% del coste del trabajador	1707.75
Coste total		9811.75

Tabla 15: Costes iniciales

Costes reales:

Concepto	Descripción	Importe en Euros
Costes Total Presupuestado		9811.75
Coste Real	(141 días x 6 h/día * 11,5 euros/día)+1273 de equipamiento	11002
Diferencia de Costes	9811.75-11002	-1190.25

Tabla 16: Costes reales

Aunque se aplicó un ajuste para prevenir el riesgo del aumento de horas que podía existir, no ha sido suficiente y nos hemos encontrado con incremento de 1190,25 euros por encima de lo presupuestado.

Capítulo 5

Requisitos

5.1. Requisitos Funcionales

A continuación se declaran los requisitos funcionales que se han identificado. La aplicación desarrollada debe proporcionar estos requisitos como funcionalidades del sistema. Estos requisitos se han desarrollado usando la herramienta REM 1.2.2.

FRQ-0001	Crear juego
Descripción	El sistema deberá permitir al usuario crear un juego introduciendo las localizaciones a localizar

FRQ-0002	Mostrar niveles de juego
Descripción	El sistema deberá mostrar los niveles de juego existentes.

FRQ-0003	Mostrar las localizaciones
Descripción	El sistema deberá mostrar al usuario las localizaciones a localizar en el mapa.

FRQ-0004	Realizar pregunta
-----------------	--------------------------

Descripción	El sistema deberá pedirle al usuario que localice un punto determinado en el mapa.
--------------------	--

FRQ-0005	Comprobar respuesta
Descripción	El sistema deberá comprobar que la localización marcada por el usuario es la correcta.

FRQ-0006	Dar puntuación
Descripción	El sistema deberá dar una puntuación según las preguntas acertadas por el usuario.

FRQ-0007	Importar un Juego
Descripción	El sistema deberá permitir al usuario importar un juego determinado.

FRQ-0008	Exportar un Juego
Descripción	El sistema deberá permitir al usuario exporta el juego.

5.2. Requisitos No Funcionales

A continuación, se declaran los requisitos no funcionales del sistema, estos se refieren a todos los requisitos de la aplicación que no describen información a guardar, ni funciones a realizar, pero que hacen posible su buen funcionamiento. Estos requisitos se han desarrollado usando la herramienta REM 1.2.2.

NFR-0001	Sistema operativo
Descripción	El sistema deberá funcionar en Smartphone o Tablet con sistema operativo Android 4.1 (Jelly Bean) o superior.

NFR-0002	Conexión a internet
Descripción	El sistema deberá tener conexión a internet para poder usar las funcionalidades de geolocalización.

NFR-0003	Tiempo de aprendizaje
Descripción	El sistema deberá permitir que un usuario medio de Android sea capaz de crear un juego en menos de 5 minutos, en más del 90 % de los casos.

NFR-0004	Seguridad de datos
Descripción	El sistema deberá trabajar con datos que no sean de carácter personal ni confidenciales.

NFR-0005	Espacio máximo requerido
Descripción	El sistema deberá de ocupar un espacio máximo de 20 MB con la base de datos vacía. Una vez iniciado el juego, se podrá añadir tantos puntos a la base de datos como espacio disponible tenga el dispositivo.

NFR-0006	Características del terminal
Descripción	El sistema deberá poder ejecutarse en el 80% de Smartphone Android de gama baja.

NFR-0007	Manual de usuario
Descripción	Se deberá proporcionar un manual de usuario donde se explique detalladamente el uso de la aplicación.

5.3. Requisitos de Información

A continuación se describen los requisitos de información que deben ser almacenados por el sistema. Uno de los aspectos más importante de un sistema informático es la información que este gestiona. Estos requisitos se han desarrollado usando la herramienta REM 1.2.2.

IRQ-0001	Localización
Descripción	El sistema deberá almacenar la información correspondiente a cada localización introducida.
Datos específicos	Nombre de la localización. Provincia Comunidad Coordenadas

5.4. Casos de Uso

En el siguiente apartado se describen los casos de usos, que no son más que los pasos que hay que seguir para llevar a cabo un proceso determinado.

UC-0001		Crear juego	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario selecciona "Crear Juego".		
Precondición			
Secuencia normal	Paso	Acción	
	1	El actor Usuario (ACT-0001) solicita al sistema crear un juego	
	2	El sistema le pide al usuario que introduzca el nombre de la localización que desea añadir.	
	3	El actor Usuario (ACT-0001) introduce el nombre de la localización a añadir.	
	4	El sistema localiza la localización introducida y obtiene las coordenadas	
	5	El sistema muestra la localización en el mapa.	
	6	El actor Usuario (ACT-0001) confirma que es la localización que desea añadir.	
	7	El sistema verifica que no existe la localización en la base de datos.	
	8	El sistema añade la localización al juego que se ha creado	
	9	El actor Usuario (ACT-0001) solicita salir.	
	10	El sistema guarda el nuevo juego.	
Postcondición	Se ha creado un nuevo juego		
Excepciones	Paso	Acción	
	6	Si el usuario no confirma que es la localización que quería añadir, el sistema regresa al paso 3, a continuación este caso de uso continúa.	
	7	Si existe la localización en la base de datos, el sistema actualiza la localización con el identificador del juego que se está creando actualmente, a continuación este caso de uso continúa.	
	8	Si el usuario desea añadir más localizaciones, el sistema regresa al paso 1, a continuación este caso de uso continúa.	

UC-0002		Comprobar respuesta	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario selecciona un punto de los situados en el mapa.		
Precondición	caso de uso 0003		
Secuencia normal	Paso	Acción	
	1	El sistema comprobará que la respuesta dada por el usuario es correcta	
	2	El sistema aumenta la puntuación del usuario.	
	3	El sistema comprueba que no se han realizado todas las preguntas y vuelve al paso 4 del UC-0003	
Excepciones	Paso	Acción	
	1	Si la respuesta dada por el usuario no es correcta, el sistema no suma puntos a la puntuación del usuario, a continuación este caso de uso queda sin efecto	
	3	Si se ha llegado al final del juego, el sistema muestra las puntuaciones alcanzadas. A continuación, este caso de uso queda sin efecto	

UC-0003		Jugar	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario inicializa el juego		
Precondición	Debe de existir un juego actual ya sea aplicando el caso de uso UC-0001 o con UC-0008.		
Secuencia normal	Paso	Acción	
	1	El actor Usuario (ACT-0001) solicita jugar.	
	2	El sistema le pregunta al usuario por una localización.	
	3	El actor Usuario (ACT-0001) introduce la respuesta y a continuación se realiza el UC-0002.	

UC-0004		Editar Juego	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario solicita Editar el Juego.		
Precondición	Debe de existir un juego actual ya sea aplicando el caso de uso UC-0001 o con UC-0008.		
Secuencia normal	Paso	Acción	
	1	El actor Usuario (ACT-0001) añade una localización.	
	2	El sistema guarda la nueva configuración del juego.	
Postcondición	El juego se ha editado.		
Excepciones	Paso	Acción	
	1	Si el usuario elimina una localización, a continuación este caso de uso continúa.	

UC-0005		Elegir Filtro	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario solicita al sistema Elegir Filtro.		
Secuencia normal	Paso	Acción	
	1	El sistema muestra las opciones de filtro.	
	2	El actor Usuario (ACT-0001) selecciona un filtro.	
	3	El sistema guarda el filtro seleccionado.	

UC-0006		Seleccionar Modo de Juego	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario solicita al sistema elegir un modo de juego.		
Secuencia normal	Paso	Acción	
	2	El sistema muestra los diferentes modos de juego.	
	3	El actor Usuario (ACT-0001) selecciona un modo de juego.	
	4	El sistema guarda el modo de juego seleccionado.	

UC-0007	Elegir Dificultad de Juego	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario solicita al sistema elegir la dificultad del juego.	
Secuencia normal	Paso	Acción
	2	El sistema muestra los diferentes niveles de juego.
	3	El actor Usuario (ACT-0001) selecciona un nivel de juego.
	4	El sistema guarda el nivel de juego seleccionado.

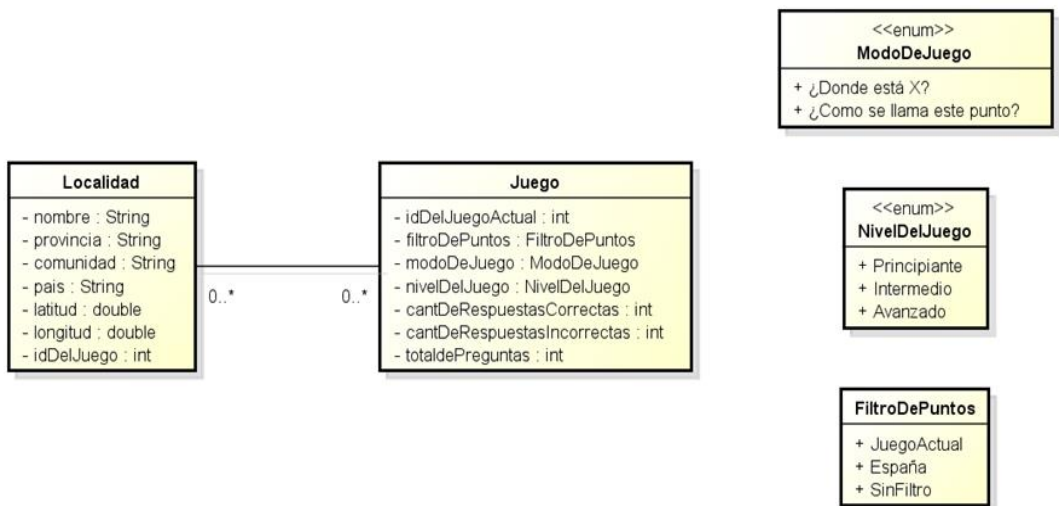
UC-0008	Importar Juego	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso	
Secuencia normal	Paso	Acción
	1	El actor Usuario (ACT-0001) solicita al sistema importar un juego.
	2	El sistema importa el juego seleccionado.

UC-0009	Exportar Juego	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso	
Precondición	Debe de existir un juego actual ya sea aplicando el caso de uso UC-0001 o con UC-0008.	
Secuencia normal	Paso	Acción
	1	El actor Usuario (ACT-0001) solicita al sistema exportar un juego.
	2	El sistema exporta el juego.

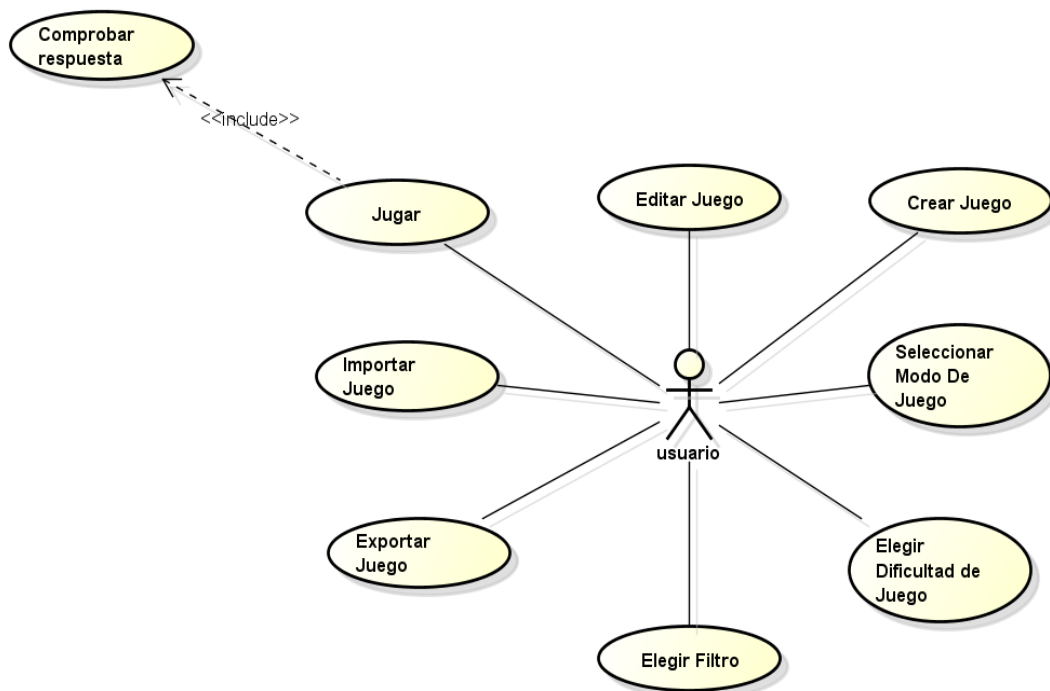
Capítulo 6

Análisis

6.1. Modelo del Dominio

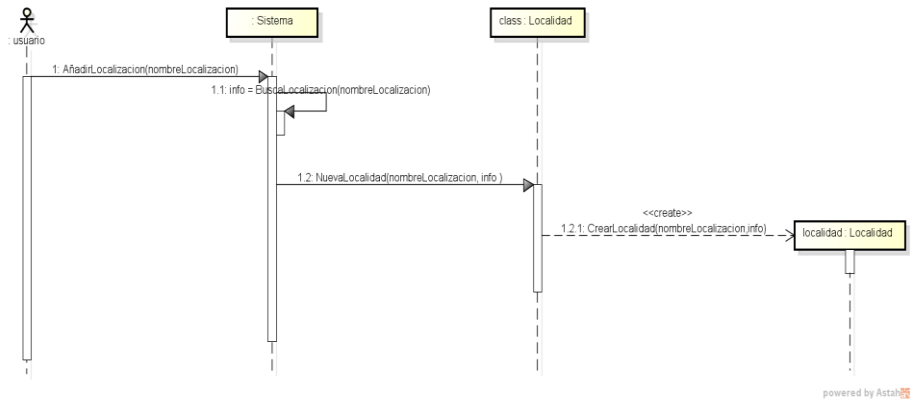


6.1. Diagrama de Casos de Uso

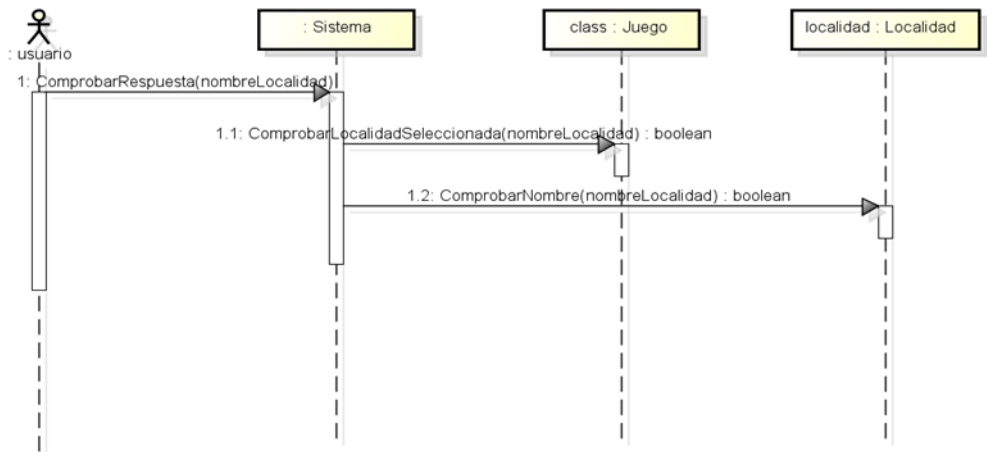


6.1. Diagrama de Secuencias

UC-0001 Crear Juego

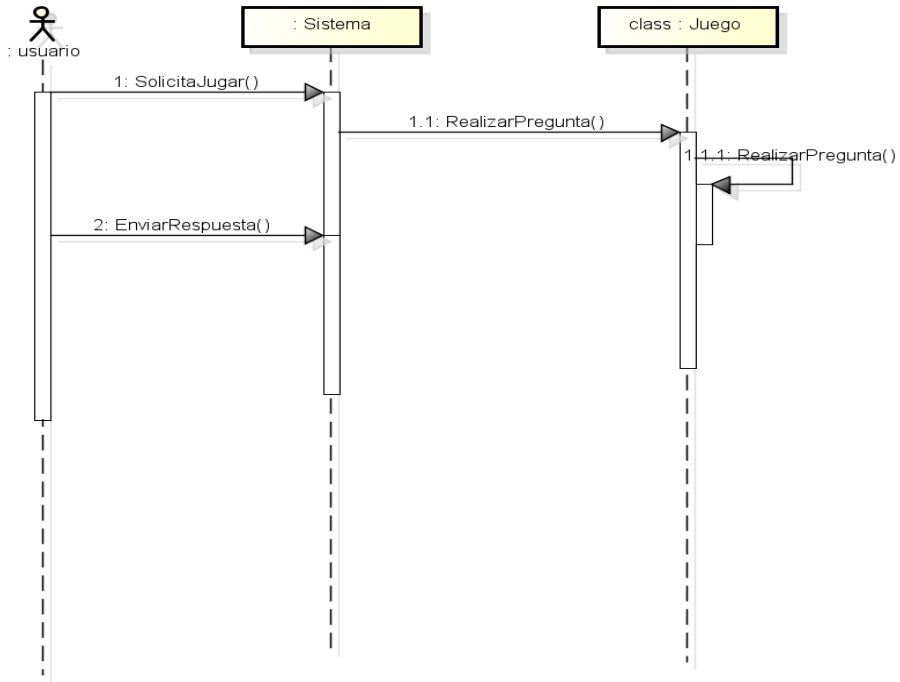


CU-0002 Comprobar Respuesta

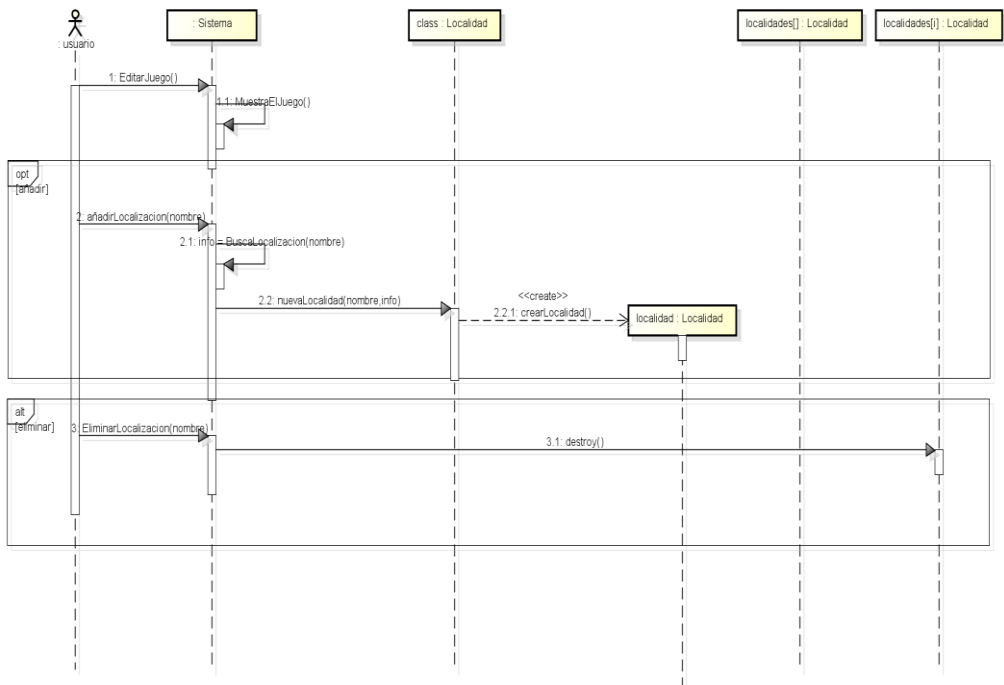


Capítulo 6: Análisis

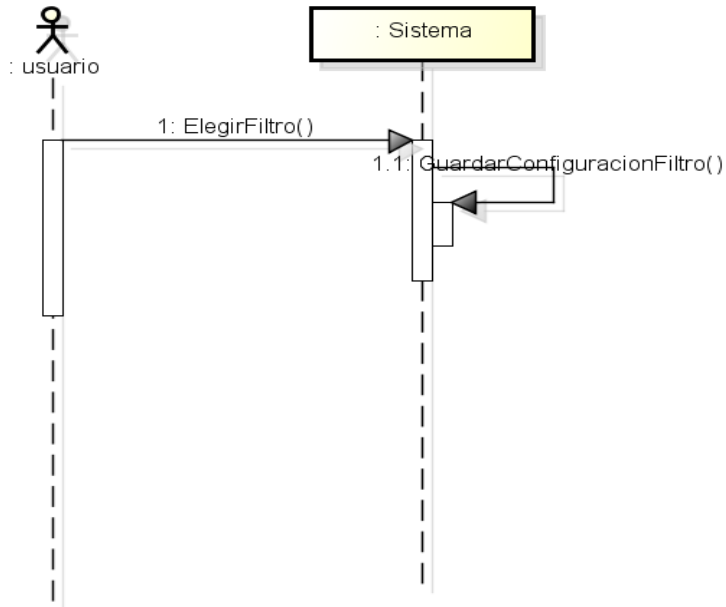
UC-0003 Jugar



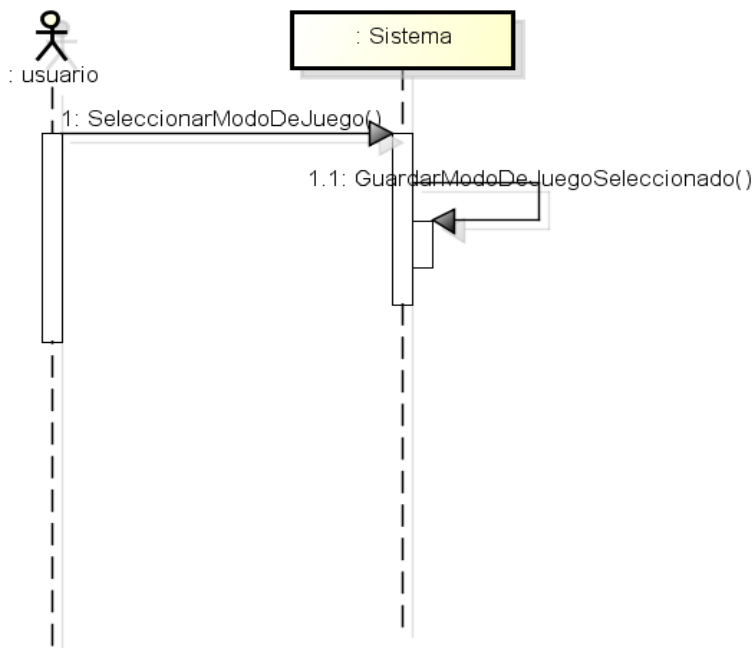
UC-0004 Editar Juego



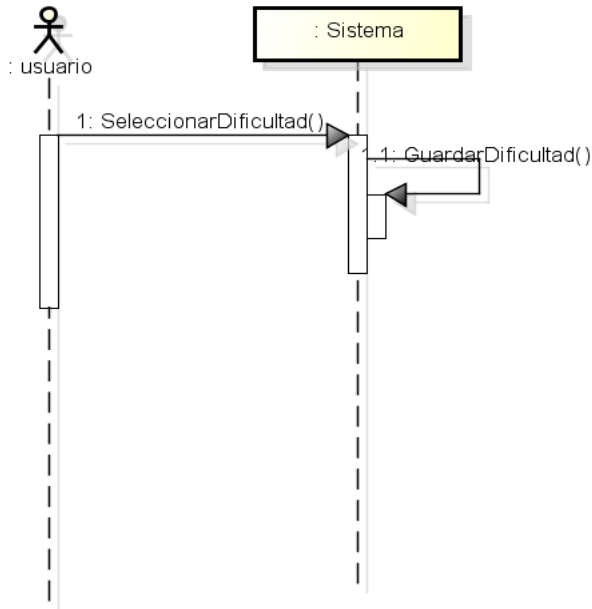
UC-0005 Elegir Filtro



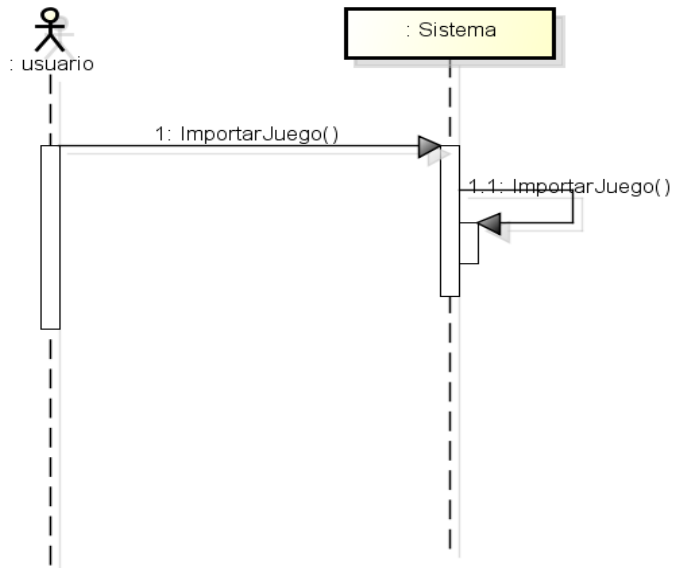
UC-0006 Seleccionar Modo De Juego



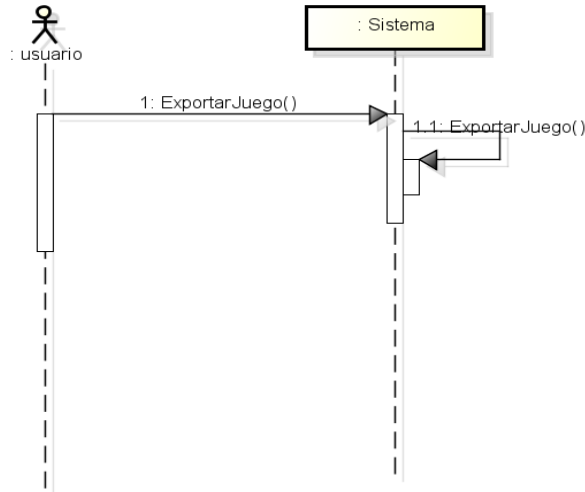
UC-0007-Elegir Dificultad Del Juego



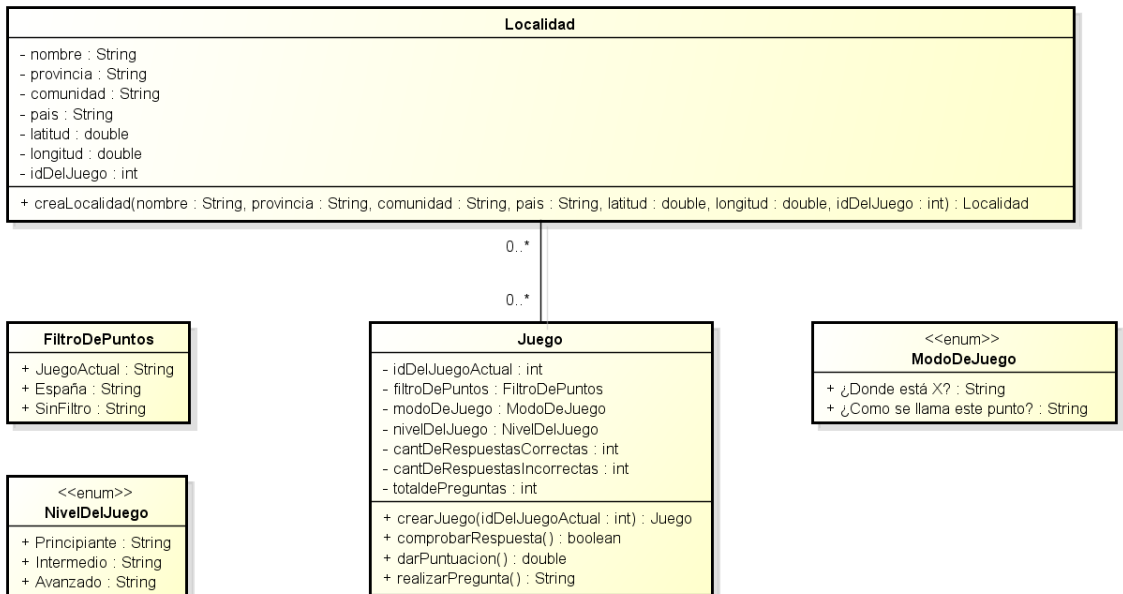
UC-0008 Importar Juego



UC-0009 Exportar Juego



6.1. Diagrama de Clases



Capítulo 7

Diseño

7.1. Estructura de Diseño

En este capítulo se detallaran las metodologías de diseño que se tendrán en cuenta durante la etapa de implementación del proyecto. El objetivo principal de este, es construir una aplicación móvil de calidad, que sea capaz de adaptarse a futuras mejoras. También se explicaran algunos aspectos a seguir como los patrones de arquitectura, el modelo de datos y diagramas de secuencias.

7.2. Descripción de la arquitectura

7.2.1.MVC

El objetivo del MVC consiste en separar los datos de una aplicación, la interfaz de usuario y la lógica de negocio en tres componentes distintos que se relacionarán.

Modelo: Es la clase que esta relacionadas con el modelo de los datos de la base de datos. El modelo gestiona también el acceso a dicha información, aquí se encuentran métodos como insertar, modificar o eliminar. Por ejemplo, si tenemos un sistema que almacena usuarios cuando estos se registran, entonces tendríamos en el paquete de modelo la clase usuario, que se correspondería al usuario de la base de datos. Las peticiones de acceso o manipulación de la información llegan a través del controlador, de esta manera separaríamos el modelo de datos de la lógica de negocio y de la vista.

Vista: Sera la parte de la aplicación que tendrá una relación directa con el usuario. La vista será la encargada de mostrar y capturar los datos del usuario. Estos datos serán recibidos por el controlador que será el encargado de procesarlos.

Controlador: Es la clase que responderá a las acciones y eventos de la aplicación. Sirve de intermediario entre el modelo y la interfaz de usuario, será el encargado de pedirle los datos al modelo, procesar esa información y enviársela a la vista. Un controlador también puede interactuar con otros controladores.

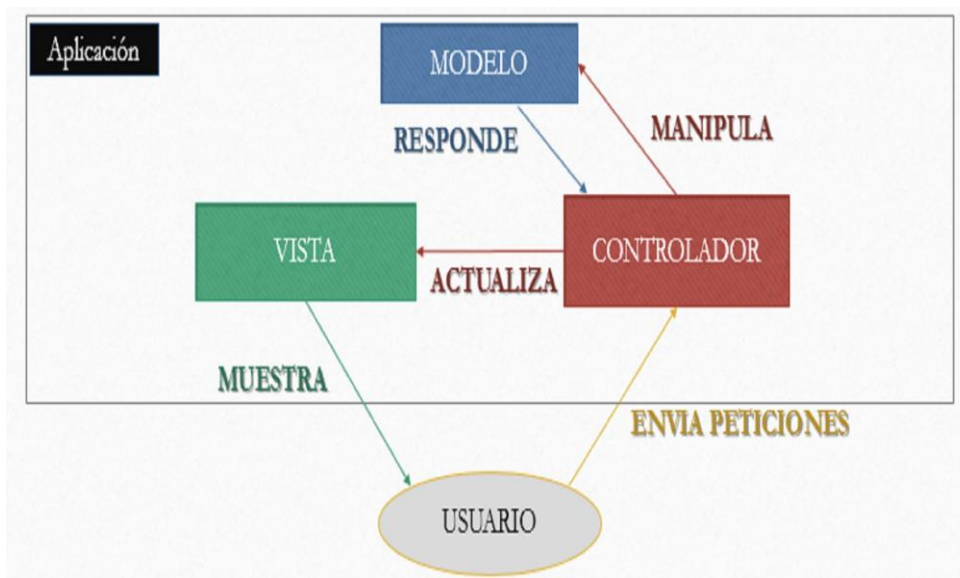


Figura 10: Modelo Vista Controlador

(Fuente [17])

7.2.2. Arquitectura global del sistema

La estructura de un proyecto en Android está compuesta por paquetes que contienen actividades, clases java, layouts y recursos en formato xml o html. Si tomamos las clases java como posibles modelos, las actividades como controladores y los layout como la vista, podríamos entonces aplicar el MVC. Aunque algunas fuentes identifican al MVP (una variante del MVC) como el patrón que más se ajusta al desarrollo de una aplicación en Android, se ha decidido por optar en un diseño de tipo MVC. A continuación, se muestra la estructura de paquetes de la aplicación desarrollada siguiendo el MVC.

Paquete modelo: El paquete Modelo de nuestra aplicación va a tener todas las clases relacionadas con la información que se guardara en la base de datos. Teniendo en cuenta esto, entonces este paquete deberá contener la clase Localidad, que es la que representa los puntos que busca el usuario en el mapa y que se deberán guardar.

Paquete vista: Siguiendo el MVC aquí se debería encontrar la interfaz de usuario de la aplicación. En una aplicación Android esto está compuesto principalmente por los layout. Nuestro paquete vista contendrá entonces los .xml que conforman los layout, además del mapa de la API v3 de Google Maps .En el caso del mapa, teniendo en cuenta que esto es producto ya echo, entonces, será

considerado como un Artefacto que se usara en la implementación de la interfaz gráfica. Cada vista en particular, podrá estar formada por vistas individuales como botones, entradas de textos y webView.

Paquete controlador: Si un controlador es el encargado de gestionar y responder a los eventos de la aplicación, entonces en este paquete tendremos principalmente a las actividades.

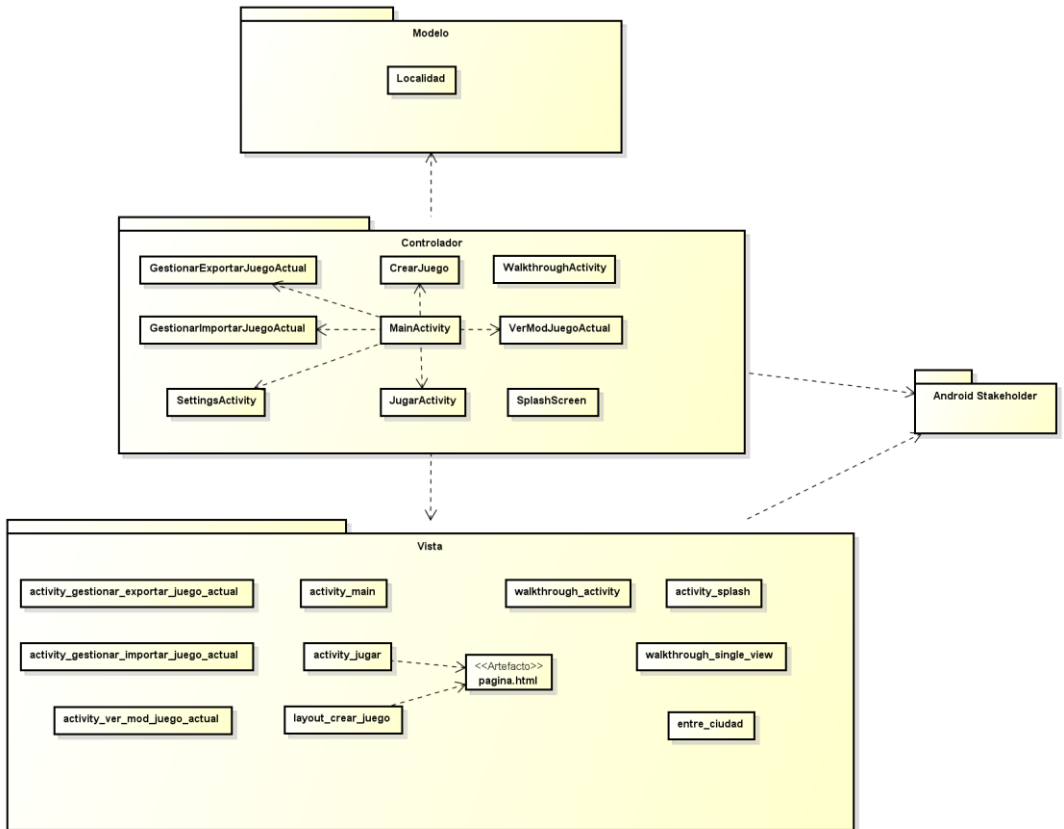


Figura 11: Arquitectura de la aplicación

7.3. Patrones de diseños utilizados

Los patrones de diseño son un conjunto de soluciones a problemas comunes en el diseño de una aplicación.

7.3.1. Patrón Singleton

El patrón de diseño Singleton, también conocido como Instancia Única, está diseñado para restringir la creación de objetos. El objetivo principal de este patrón es que solo exista una única instancia de esa clase en toda la aplicación, de esta manera tendríamos solo un punto de acceso global al objeto creado desde cualquier parte del código.

Importancia:

En ocasiones es necesario no tener más de una instancia de una clase, como por ejemplo, cuando tenemos un historial de datos que será mostrado al usuario y este es común para todos, o si tuviéramos una clase Conexión que usa toda la aplicación, restringiendo la creación de instancias de este tipo de clase podemos evitar posibles problemas.

Funcionamiento:

- Oculta el constructor para que los clientes no puedan crear más de una instancia.
- Declarar una variable privada en la clase Singleton que contenga la referencia a la instancia única que queremos gestionar.
- La clase Singleton contendrá un método de acceso a la única instancia gestionada por el Singleton, los clientes podrán acceder a dicha instancia desde ese método de acceso.

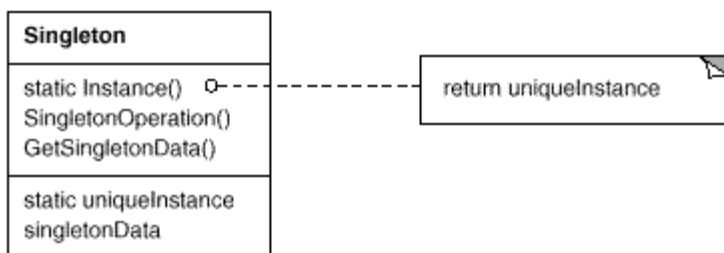


Figura 12: Diagrama OMT de Singleton

(Fuente [18])

Este patrón de diseño será utilizado en el proyecto para la clase ConnectionPool que será la encargada de gestionar las conexiones a la base de datos de la aplicación.

7.3.2.Iterator

Este patrón de diseño, permite acceder secuencialmente a una colección de objetos sin exponer su representación interna.

Se utilizara para recorrer la lista de localidades, cuando no sea necesario conocer la estructura interna de dicha clase, o cuando sea necesario diferentes tipos de recorridos.

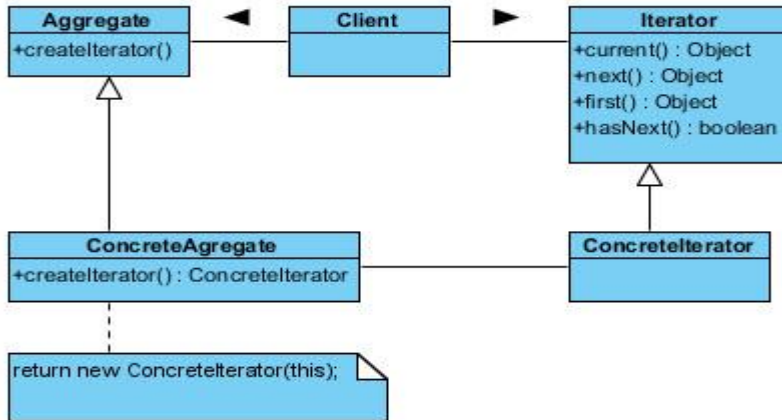


Figura 13: Estructura Iterator

(Fuente [19])

7.1. Diagrama de clases por funcionalidad

A continuación, se muestran los diagramas de clases relacionados con el diseño de la aplicación. Para una mejor visualización estos diagramas se han separado por funcionalidad de la aplicación.

7.1.1.1. Jugar

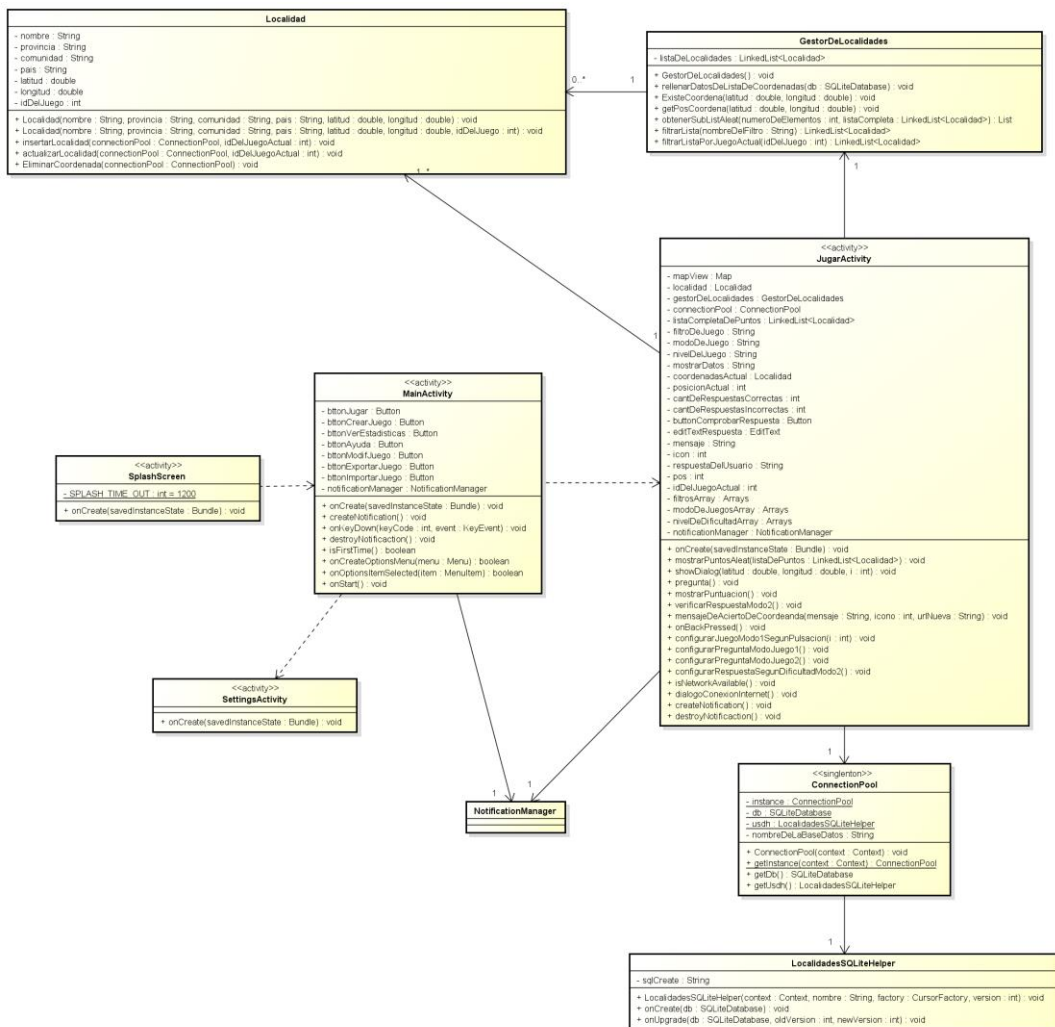


Figura 14: Diagrama de clases Jugar

7.1.1. Crear Juego

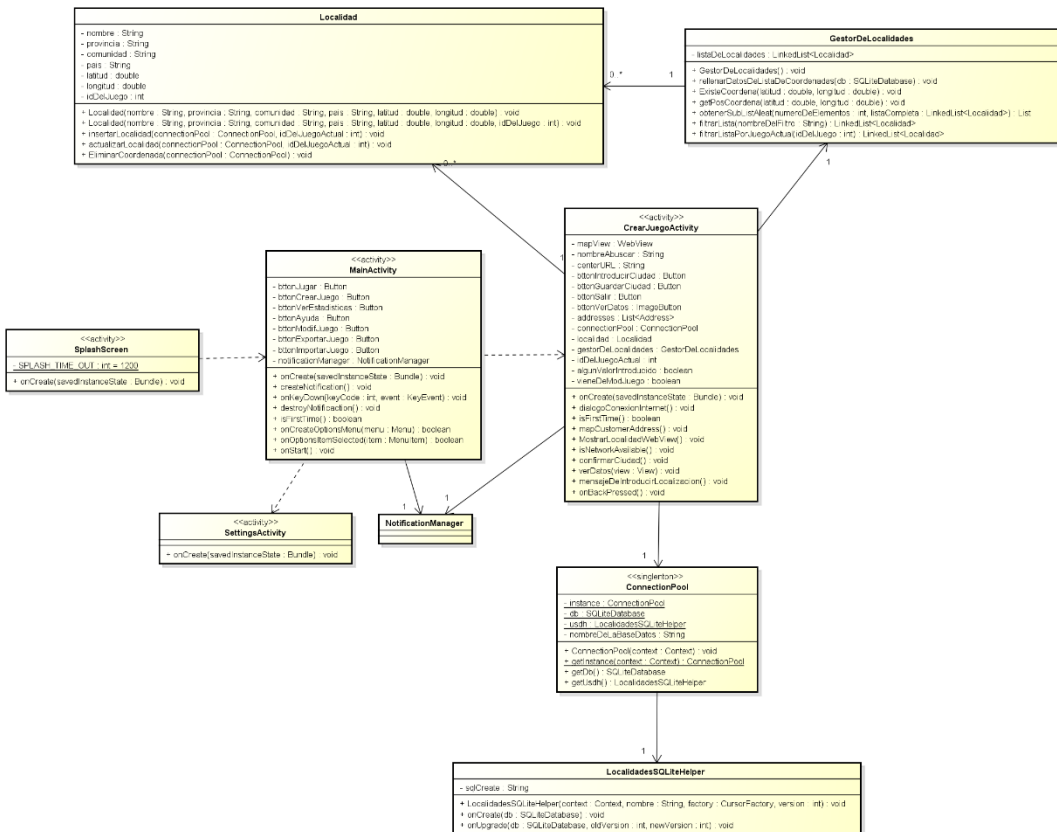


Figura 15: Diagrama de clases Crear Juego

7.1.1. Editar un Juego

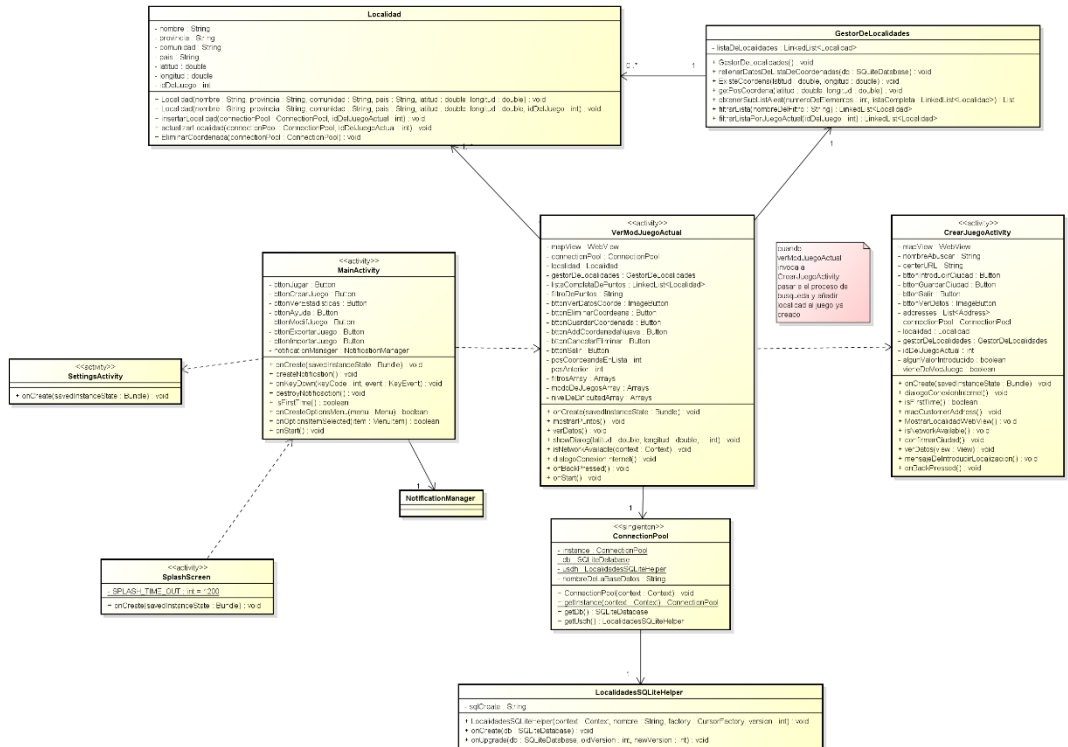


Figura 16: Diagrama de clases Editar Juego

7.1.1. Importar Juego

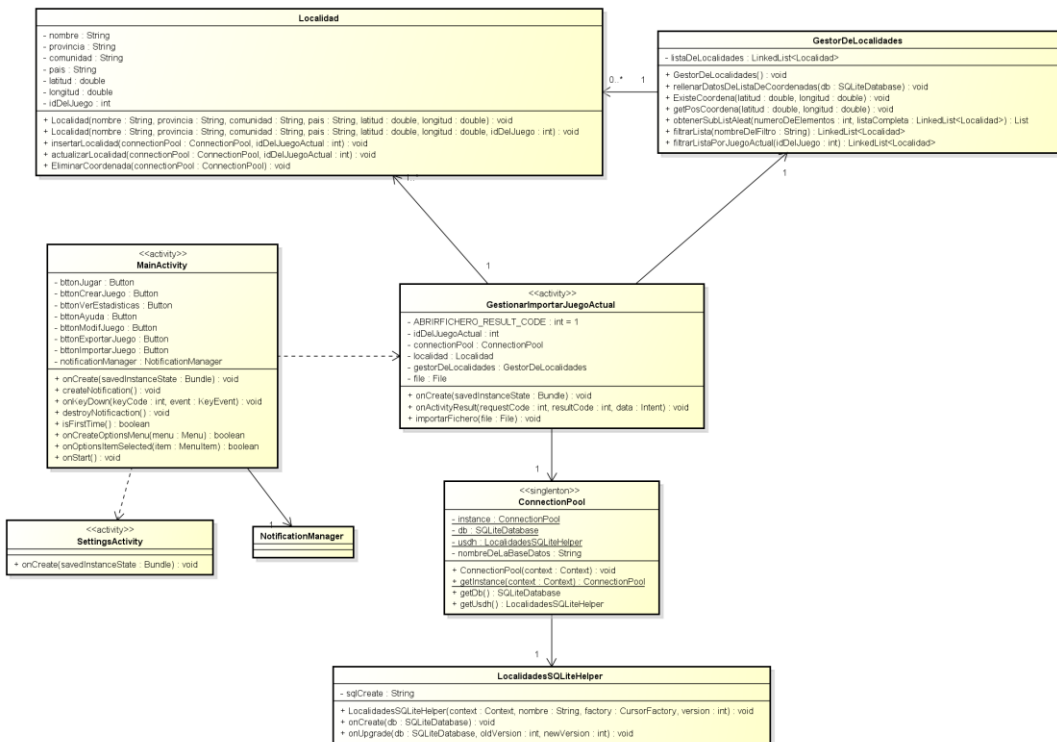


Figura 17: Diagrama de clases Importar un Juego

powered by ActOn

7.1.1. Exportar Juego

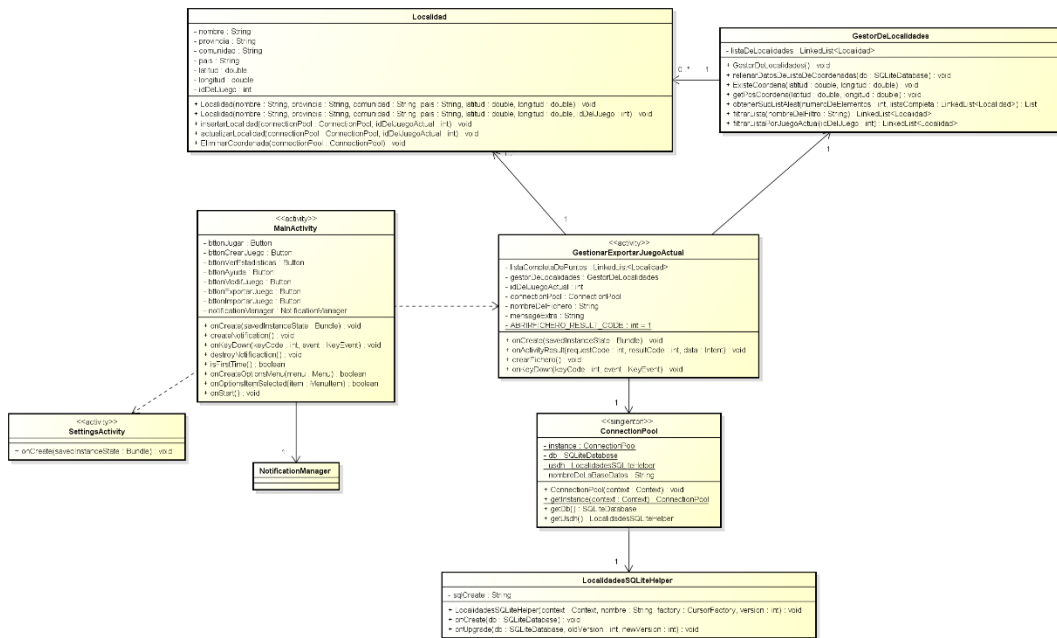


Figura 18: Diagrama de clases Exportar un Juego

7.1.2. Ayuda

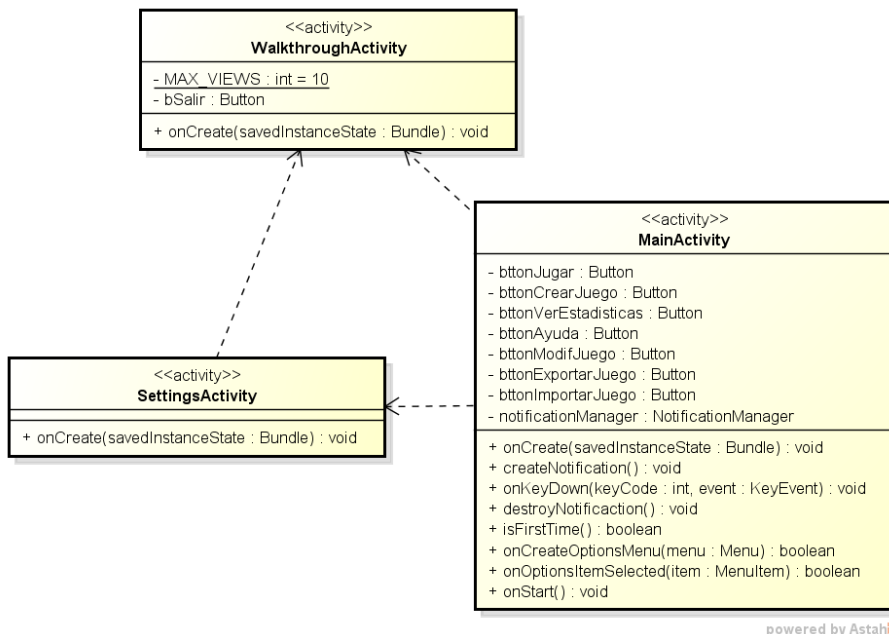
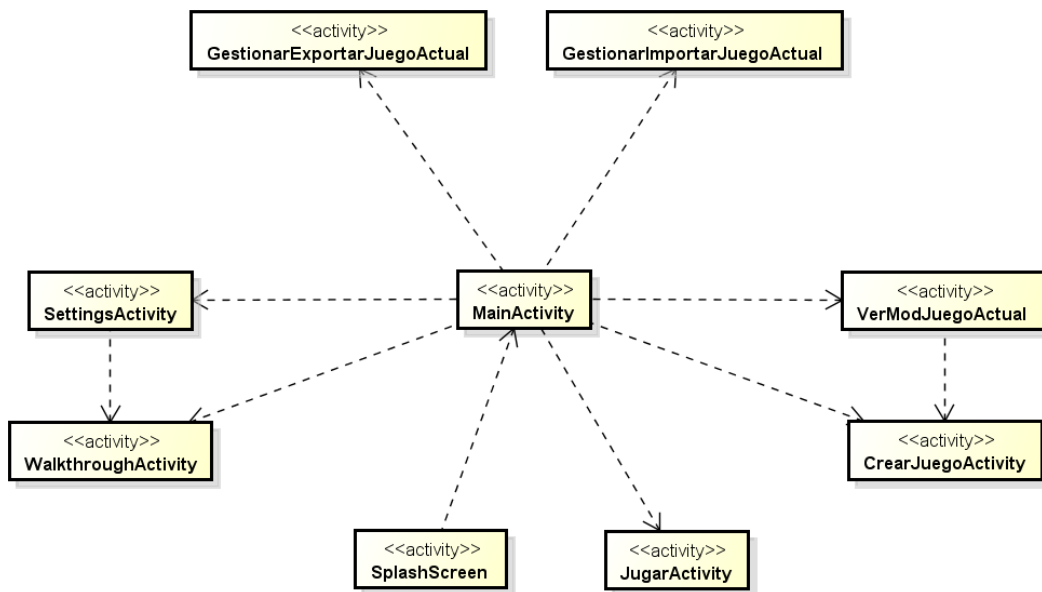


Figura 19: Diagrama de clases Ayuda

7.1.1. Relación entre las Activities



powered by Astah

Figura 20: Diagrama de relación entre actividades

7.2. Modelo de datos

Un modelo de datos para la base de datos incluye una serie de conceptos como entidades, atributos y relaciones, los cuales son utilizados para la descripción de una base de datos. Normalmente estos modelos de datos poseen operaciones básicas para realizar consultas, actualizaciones e inserciones la base de datos.

En este apartado se describirá el modelo de datos que será usado en la aplicación.

7.2.1. Base de datos SQLite

El tratamiento y gestión de la base de datos correrá a cargo de SQLite, la cual es una herramienta de dominio público que se integra muy bien para las necesidades de esta aplicación.

La app a desarrollar no contará con un servidor de datos, ya que estos no son comunes para todos los usuarios, por lo cual tendrán que estar almacenados en cada dispositivo en particular. SQLite se acopla muy bien a esta necesidad al ser independiente, no se necesita de un servidor para la base de datos, sino que se realizan llamadas a funciones o subrutinas que están incluidas en la propia librería, los datos son almacenados en un solo fichero en el dispositivo.

La herramienta SQLite presenta un conjunto de ventajas que facilitan la creación de una base de datos:

- La plataforma de Android incluye la herramienta de SQLite para el tratamiento de datos.
- No requiere configuración alguna
- En su versión 3 permite bases de datos de hasta 2 Terabytes
- Acceso mucho más rápido.
- Único archivo de base de datos.
- Más de 2/3 del código de SQLite está dedicado exclusivamente a la prueba y verificación favoreciendo con esto la seguridad de los datos.
- Se puede exportar fácilmente en caso de que esto fuera necesario.

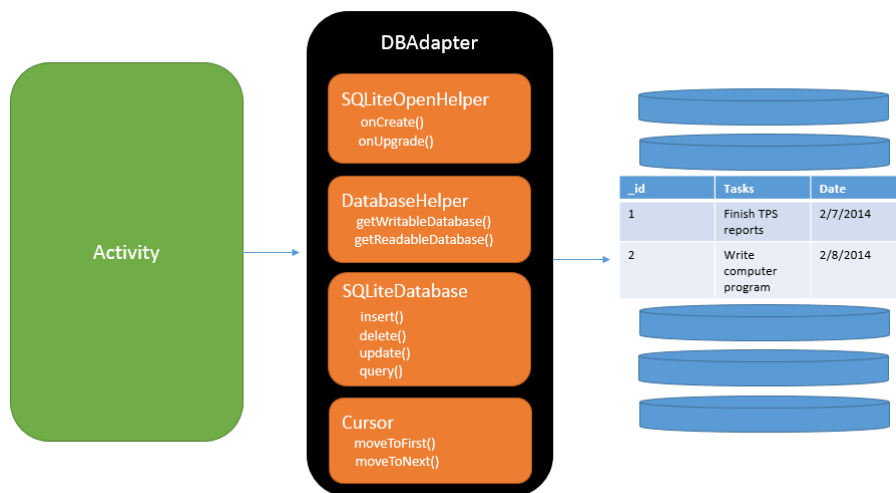


Figura 21: Base de datos SQLite
(Fuente [20])

7.2.2. Modelo de datos de la App

Se utilizara una única tabla que estará almacenada de manera local.

Localidad	
- <<PK>>	latitud : double
- <<PK>>	longitud : double
-	nombre : String
-	provincia : String
-	comunidad : String
-	pais : String
-	idDelJuego : int

powered by Astah

Figura 22: Tabla de la base de datos

7.1. Diagramas de secuencias

A continuación, se mostraran algunos de los diagramas de secuencias, que describen como interactúa el sistema ante algunas peticiones del usuario.

7.1.1. Exportar un Juego

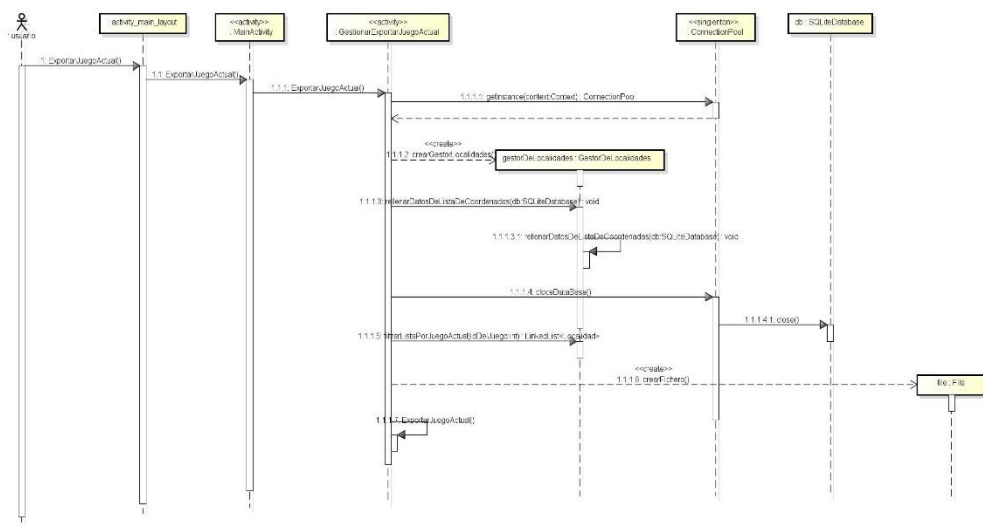
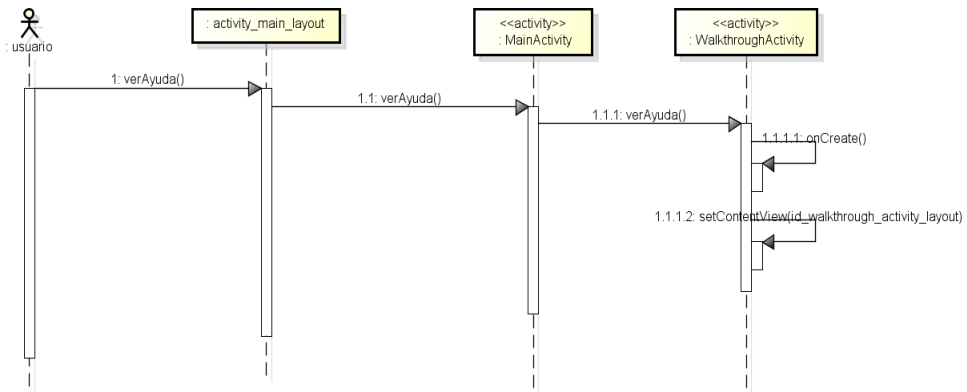


Figura 23: Diagrama de secuencia Importar Juego

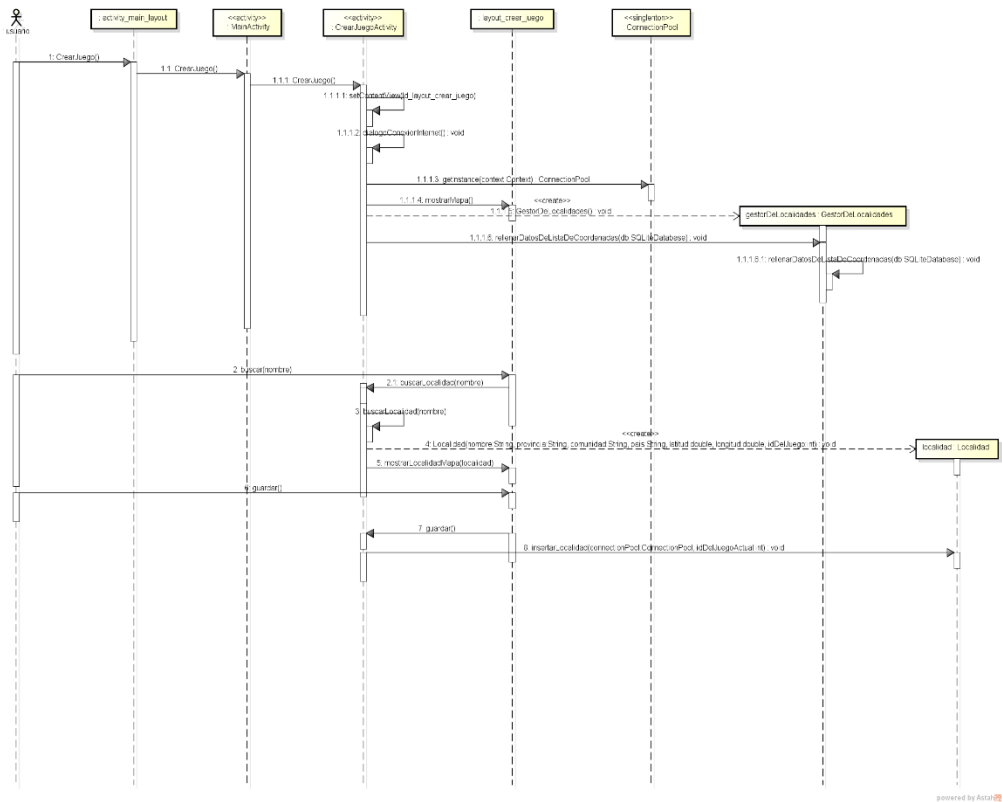
7.1.2. Mostrar Ayuda



powered by Astah

Figura 24: Diagrama de secuencia Mostrar Ayuda

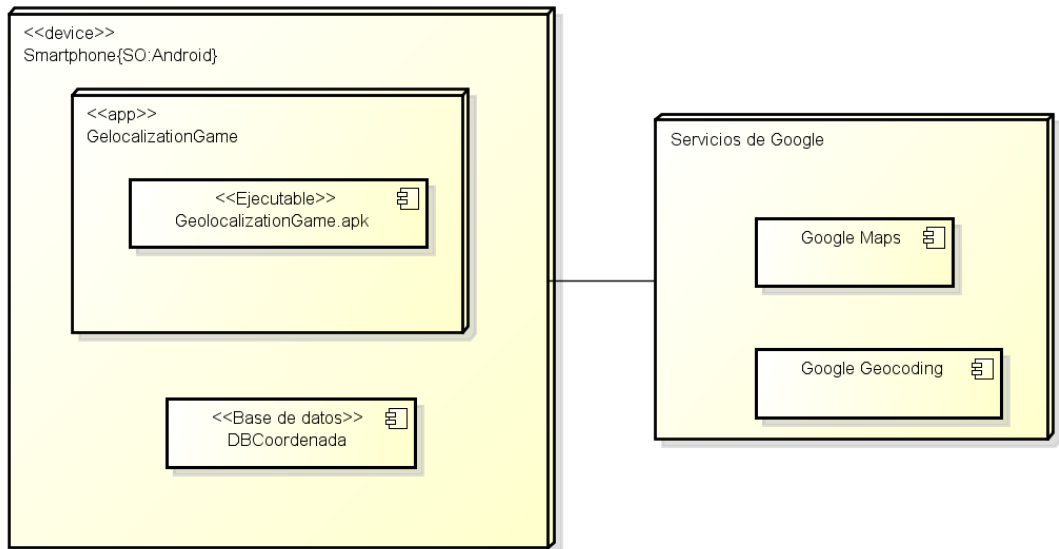
7.1.1. Crear un Juego, añadir una localidad



powered by Astah

Figura 25: Diagrama de secuencia Crear Juego, añadir una localidad

7.2. Diagrama de Despliegue



powered by Astah

Figura 26: Diagrama de despliegue

Como se puede observar en diagrama anterior, la aplicación estará compuesta por una arquitectura cliente-servidor, donde el cliente será el dispositivo que contendrá la aplicación y la base de datos y el servidor será los servicios de google que se utilizaran.

Capítulo 8

Implementación

8.1. Introducción

En el siguiente capítulo, se explicara los pasos necesarios para el uso de la API v3 de Google Maps en una aplicación Android. Debido a que se ha necesitado usar una programación hibrida para el uso de la API v3, se han necesitado unos mecanismos de comunicación entre la plataforma Android y la página HTML que contenía el mapa. Estos mecanismos de comunicación entre los diferentes lenguajes, también serán explicados y detallados.

8.2. Utilización de la API v3 de Google Maps en Android

La API v3 de Google Maps esta implementada en Java Script ,por la tanto si queremos introducirla en una aplicación Android tendremos que crear una página HTML que contenga el mapa e introducirla en un WebView en nuestra aplicación.

8.2.1. WebView

Un WebView es un componente que proporciona el SDK de Android que incluye un motor de renderizado de HTML y un intérprete de JavaScript .Las páginas web a mostrar, pueden ser cargadas tanto obteniéndolas mediante una URL o cargando una página HTML desde una actividad. WebView también proporciona una serie de métodos que entre otras cosas permiten la gestión de cookies y el procesamiento de una serie de errores. A continuación, se muestra un fragmento del código utilizado en el TFG para crear el WebView.

Capítulo 8: Implementación

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="end"
    android:orientation="vertical" >
```

```
<FrameLayout
    android:id="@+id/fram_map"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >
```

```
<WebView
    android:id="@+id/mapView"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
```

```
</FrameLayout>
```

```
<FrameLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_centerInParent="true" >
```

```
<GridLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:columnCount="1"
    android:orientation="horizontal" >
```

```
<Button
    android:id="@+id/buttonIntroducirCiudad"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_column="0"
    android:layout_gravity="left|bottom"
    android:layout_row="0"
    android:text="@string/add"
    android:drawableTop="@drawable/search" />
```

```
</GridLayout>
```

```
</FrameLayout>
```

```
</RelativeLayout>
```

A pesar de que el WebView también proporciona los mecanismos necesarios para crear las interfaces grafica de la app, este se ha incluido en un FrameLayout (Fragmento).El objetivo principal es que todos los componentes que se añadan a la vista del mapa (botones, campos de textos, imágenes, etc.), sean totalmente de la interfaz de Android, de esta manera se conseguiría una interfaz más nativa.

8.2.1.Creando la página HTML que contendrá el mapa

A continuación, se creará una página web básica donde se cargue el mapa de la Api v3 de Google Maps.

Lo primero que debemos hacer es crear una página HTML5 en la cual podamos introducir el mapa:

```
<!DOCTYPE html>
<html>

<head>
<title>Página básica </title>
<meta name="viewport" content="initial-scale=1.0, user-scalable=no">
<meta charset="utf-8">
</head>

<body>
<div id="mapa"></div>
</body>

</html>
```

El siguiente paso es agregar el siguiente script que permite incluir una referencia de la API de Google Maps. La versión 3 del API de Google Maps JavaScript no requiere una clave de API para funcionar correctamente, por lo tanto no la vamos a añadir. El parámetro sensor se utiliza para indicar que vamos a usar algún tipo de sensor, por ejemplo para localizar la posición del usuario mediante GPS. Como en la app desarrollada no vamos a usar ningún sensor de momento, ponemos el valor como false.

```
<script
  type="text/javascript"
  src="http://maps.googleapis.com/maps/api/js?sensor=false">
</script>
```

Capítulo 8: Implementación

Un aspecto importante para que el mapa se renderize es el siguiente código que debemos de incluir en el head:

```
<style>
  html, body, #mapa {
    margin: 0;
    padding: 0;
    height: 100%;
  }
</style>
```

Luego instanciamos e inicializamos el mapa. La función initialize() será la primera que ejecutemos una vez cargada la página, por esta razón en esta función configuramos el aspecto que tendrá nuestro mapa , por ejemplo el zoom , donde estará centrado el mapa, el tipo de carretera ,etc. También podríamos crear varios estilos del mapa y cargarlos según se vayan necesitando, para esto usaríamos map.setOptions({styles: styles1}). Una vez configurado el aspecto que tendrá nuestro mapa, entonces le damos valor a la variable map pasándoles la configuración anteriormente echa. Para que la función initialize() se cargue al cargar la página debemos añadirle la última línea del código siguiente.

```
<script type="text/javascript">
  var map;

  function initialize() {
    var mapOptions = {
      zoom: 8,
      center: new google.maps.LatLng(40.41,-3.70),
      mapTypeId: google.maps.MapTypeId.ROADMAP
    };
    map = new google.maps.Map(document.getElementById('mapa'), mapOptions);
  }

  google.maps.event.addListener(window, 'load', initialize);
</script>
```

Uniéndolo todo la página básica .html con el mapa a cargar debería de tener este aspecto:

```
<!DOCTYPE html>
<html>
  <head>

    <title>página simple con mapa v3 </title>
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no">
    <meta charset="utf-8">

    <style>
      html, body, #mapa {
        margin: 0;
        padding: 0;
        height: 100%;
      }
    </style>

    <script src="https://maps.googleapis.com/maps/api/js?v=3.exp&sensor=false"></script>

    <script>
      var map;
      var myPos;
      function initialize() {
        var mapOptions = {
          zoom: 8,
          center: new google.maps.LatLng(40.41,-3.70),
          mapTypeId: google.maps.MapTypeId.ROADMAP
        };
        map = new google.maps.Map(document.getElementById('mapa'), mapOptions);
      }
      google.maps.event.addDomListener(window, 'load', initialize);
    </script>
  </head>

  <body>
    <div id="mapa"></div>
  </body>

</html>
```

8.2.1. Cargando el mapa en Android

Una vez creada nuestra página.html con el mapa la introducimos en la carpeta assets que se encuentra en la estructura de nuestro proyecto Android. A continuación, en la actividad que queremos que despliegue el WebView con el mapa escribimos el siguiente código básico:

```
//obtenemos el WebView donde queremos mostrar el mapa.
mapView = (WebView) findViewById(R.id.mapViewJugar);

//Lo habilitamos para que se pueda ejecutar código JavaScript
mapView.getSettings().setJavaScriptEnabled(true);

//guardamos la URL con la página que contiene el mapa, la que está en la carpeta assets.
final String MAP_URL = "file:///android_asset/página.html";

//Cargamos el mapa en el WebView pasándole la URL.
mapView.loadUrl(MAP_URL);
```

8.3. Vinculando código de JavaScript a Android

La comunicación entre la página .html creada anteriormente y el código de Android de nuestra aplicación, puede ser de mucha importancia si nuestro objetivo principal es tener una aplicación lo más nativa posible y una interfaz totalmente de Android.

En el TFG esta comunicación ha sido necesaria. A continuación, mostramos un fragmento del código utilizado en la clase JugarActivity.java.

```
@JavascriptInterface
public void showDialog(double latitud, double longitud, int i) {

    if (modoDeJuego.equals(modoDeJuegosArray[0])){

        configurarJuegoModo1SegunPulsacion(i);

        ...

        mensajeDeAciertoDeCoordeanda(mensaje, icon, MostrarDatos);

        // Preguntar nueva localización
        pregunta();

    }
}
```

El objetivo principal del código de ejemplo, es obtener el id del marcador pulsado en el mapa y desde Android usarlo para cambiar la animación y comprobar si la respuesta seleccionada es correcta o no. Lo importante de este ejemplo, no es el código que se realiza dentro de la función sino como pasar valores de la página web hasta Android por parámetros de la función declarada.

Para vincular el código anterior con el de JavaScript que se va a ejecutar en nuestro WebView deberemos usar el método `addJavascriptInterface()` indicando la instancia de la interfaz y el nombre de dicha instancia. El código se muestra a continuación, donde `mapView` es el WebView que contiene el mapa.

```
mapView.addJavascriptInterface(JugarActivity.this, "Android");
```

Ahora desde nuestra página .html bastaría con hacer una llamada a la interfaz declarada anteriormente y pasarle los valores que necesitemos capturar. A continuación, se muestra un fragmento del código usado en la página.html del TFG donde al final se realiza la llamada a la función desarrollada en Android.

```
//sin animación para el juego
```

```
function addMarker1(latitude, longitude, i) {  
    var myLatLng = new google.maps.LatLng(latitude, longitude);  
    var marker1 = new google.maps.Marker({position: myLatLng ,map: map});  
    markersArray[i]=marker1;  
    //markersArray.push(marker1);  
    limites.extend(myLatLng);  
    map.fitBounds(limites);  
  
    //enviando a Android las coordenadas del marcador  
    google.maps.event.addListener(marker1, 'click', function() {  
        Android.showDialog(latitude,longitude,i);  
    });  
}
```

8.1. Ejecutando una función JavaScript declarada en la página HTML desde Android

A veces es necesario establecer la comunicación desde Android a JavaScript, por ejemplo para cambiar la animación de un marcador o centrar el mapa en unas coordenadas determinadas. El código que se muestra a continuación fue usado en la clase `CrearJuegoActivity.java` para centrar el mapa dado unas coordenadas.

Capítulo 8: Implementación

```
private void setupWebView(){  
  
    centerURL = "javascript:centerAt(" +localidad.getLatitude() + ","  
                +localidad.getLongitude()+  
                ")";  
    mapView.loadUrl(centerURL);  
  
    ...  
  
}
```

El código anterior es simple, guardamos en una variable de tipo String la función que queremos que se ejecute, indicándole el nombre de la función a cargar y los valores que le vamos a pasar por parámetro. Luego solo nos quedaría cargar el String en el WebView para que se ejecute la función.

Función declara en JavaScript en página.html cargada en el código anterior.

```
//centrar y borrar el mapa  
function centerAt(latitude, longitude){  
    myLatLng = new google.maps.LatLng(latitude,longitude);  
    placeMarker(myLatLng);  
}
```


Capítulo 9

Pruebas

9.1. Introducción

En la realización de un proyecto Android como en cualquier otro, la realización de pruebas en las diferentes fases de desarrollo es sumamente importante. Si un problema es encontrado a tiempo lo más probable es que este se pueda erradicar o en el peor de los casos disminuir su coste. A lo largo de la historia de la informática ha habido grandes errores de software que han ocasionado muchas pérdidas tanto materiales como humanas. Hay errores tan sencillos como un redondeo mal hecho que puede ocasionar pérdidas humanas. En la siguiente tabla, se muestran algunos de los errores informáticos más catastróficos de la historia.

Nombre	Año	Coste	Causa
Marinero sin rumbo	1962	18,5 millones de dólares.	Un programador codifico erróneamente en el software una formula manuscrita.
El patriota les falla a los soldados.	1991	28 soldados muertos y 100 heridos.	Un error de redondeo.
Mars Climate Orbiter	1999	655 millones de dólares	Un programa calculaba la distancia en unidades inglesas, mientras que otro utilizó unidades métricas.
El error en los frenos de los Toyota	2010	3 billones de dólares	Un error en el programa informático causaba un retraso en el sistema anti-bloqueo de los frenos.

Tabla 17: Algunos errores informáticos de la historia.

En este capítulo, se detallara el plan de pruebas que se ha seguido para evaluar la aplicación desarrollada. Además se mostraran los resultados obtenidos para los diferentes dispositivos en los que se han realizados dichas pruebas.

9.2. Plan de pruebas

Para la realización de las pruebas de la aplicación desarrollada, nos hemos basado en tres tipos fundamentalmente: pruebas unitarias, pruebas de integración y pruebas de sistema.

9.2.1. Pruebas unitarias

Este tipo de pruebas se utilizan principalmente para comprobar que el funcionamiento de un módulo de códigos de la aplicación, totalmente aislado, funciona correctamente. Se desarrollan principalmente en la etapa de implementación y el encargado de realizar estas pruebas es el programador, ya que es el que mejor conoce el código y sabe que es lo que debe hacer exactamente.

En la aplicación desarrollada estas pruebas han tenido lugar a medida que se realizaba cada método. El objetivo principal de esto era garantizar el correcto funcionamiento de una función antes de implementar otra, de esta manera si se encontraba un error era muy probable que solo tuviera que ver con el código nuevo que se estaba desarrollando. No obstante a esto, al realizar las pruebas de integración entre módulos de códigos se encontraron errores lo que llevó a nuevas pruebas unitarias entre los módulos ya revisados.

9.2.2. Pruebas de integración

Las pruebas de integración se realizan durante la etapa de desarrollo del software una vez concluida las pruebas unitarias. El objetivo principal de esto es comprobar que la integración entre varios módulos de códigos funciona correctamente.

9.2.3. Pruebas de sistema

Este tipo de pruebas se suelen realizar una vez concluidas las pruebas de integración. El objetivo principal es probar todo el sistema completo e integrado. Este tipo de prueba engloba otras como las pruebas funcionales, pruebas de estrés y pruebas de carga.

Las pruebas de sistemas normalmente son de caja negra, esto significa que la persona que realiza estas pruebas puede no tener acceso al código y se estudia el sistema desde el punto de vista de las entradas, salidas y respuestas que este produce.

9.3. Casos de Pruebas

A continuación, se detallarán las diferentes pruebas de sistema, guiadas por casos de uso, a los que fue sometida la aplicación. Entre otras cosas se indicará cual es el resultado esperado para cada caso de pruebas, de esta manera podemos contractarlos con los resultados obtenidos para los diferentes dispositivos. Los casos de prueba serán desarrollados tanto para el idioma español como para inglés, el resultado debe ser el mismo.

9.3.1.Creación de un juego nuevo

CP-001	Añadir una localidad nueva
Descripción	Añadir una localidad nueva cuando se está creando un juego.
Acción	Ir al menú de Crear Juego, buscar una ciudad y guardarla.
Resultado esperado	La aplicación debería crea un juego con dicha localidad añadida. Debería de mostrarse un Toast indicando que se ha guardado correctamente.

CP-002	Ver botón de Información de una localidad
Descripción	Se habilita el botón de información de la localidad buscada.
Acción	Ir al menú de Crear Juego, buscar una ciudad y ver el botón de información.
Resultado esperado	Se debería de mostrar los datos de la localidad buscada (nombre, provincia, comunidad, país).

CP-003	Añadir una localidad ya existente
Descripción	Añadir una localidad cuando esta ya existe en la base de datos
Acción	Ir al menú de Crear Juego, buscar una ciudad ya añadida y guardarla.
Resultado esperado	La localidad añadida debe tomar el id del nuevo juego y se debería de mostrar un Toats indicando que el dato se ha actualizado.

CP-004	Crear un juego
Descripción	Crear un juego nuevo y ver si se ha creado correctamente.
Acción	Ir al menú de Crear Juego, añadir varias ciudades y salir. Luego ir al menú de Jugar.
Resultado esperado	La aplicación debería de mostrar un mapa con las ciudades que hemos introducido al crear el juego nuevo.

9.3.1. Jugar

Las pruebas relacionadas con Jugar que a continuación se muestran, se realizaran para los diferentes tipos de filtros: Juego Actual, Sin filtro, España, Francia. El resultado deberá de ser el mismo variando solo los puntos que se muestran en el mapa.

CP-005	Jugar I
Descripción	Jugar con la configuración: Juego Actual, ¿Dónde está X?, Principiante.
Acción	Ir a las opciones del juego del menú principal. Seleccionar la siguiente configuración. Elegir Filtro: Juego Actual Elegir Modo de Juego: ¿Dónde está X? Elegir Dificultad del Juego: Principiante Luego seleccionar Jugar del menú principal y marcar respuestas correctas e incorrectas.
Resultado esperado	La aplicación debería de mostrar un mapa con las ciudades que hemos introducido al crear el juego nuevo. Al seleccionar una respuesta correcta: mostrara un dialogo indicando que la respuesta es correcta y marcara en el mapa el punto seleccionado como correcto. Si la respuesta seleccionada es incorrecta: mostrara un dialogo indicando que la respuesta es incorrecta y marcara este punto en el mapa como incorrecto.

CP-006	Jugar II
Descripción	Jugar con la configuración: Juego Actual, ¿Dónde está X?, Principiante. Comprobar que al finalizar el juego devuelve una lista con los errores cometidos.
Acción	Ir a las opciones del juego del menú principal. Seleccionar la siguiente configuración. Elegir Filtro: Juego Actual Elegir Modo de Juego: ¿Dónde está X? Elegir Dificultad del Juego: Principiante Luego seleccionar Jugar del menú principal y marcar respuestas correctas e incorrectas hasta que el juego finalice.
Resultado esperado	La aplicación debería de mostrar un mapa con las ciudades que hemos introducido al crear el juego nuevo. Al finalizar el juego la aplicación mostrara una lista con el nombre de las ciudades donde se han cometido errores de localización.

CP-007	Jugar III
Descripción	Jugar con la configuración: Juego Actual, ¿Dónde está X?, Principiante. Comprobar el cambio en el mapa de una localidad marcada previamente como incorrecta en una respuesta correcta.
Acción	Ir a las opciones del juego del menú principal. Seleccionar la siguiente configuración. Elegir Filtro: Juego Actual Elegir Modo de Juego: ¿Dónde está X? Elegir Dificultad del Juego: Principiante Luego seleccionar Jugar del menú principal y marcar respuestas incorrectas hasta que te pregunte el juego por una localidad la cual ya has marcado como incorrecta. Selecciona dicha localidad correctamente.
Resultado esperado	Deberá de mostrar un dialogo indicando que la respuesta es correcta y cambiara el marcador de la localidad de incorrecta a correcta.

CP-008	Volver a jugar
Descripción	Cuando se seleccione Volver a Jugar el juego debería de comenzar cambiando el orden en que se realizan las preguntas.
Acción	Seleccionar Jugar del menú principal y marcar hasta que el juego finalice. Dar a la opción de Volver a Jugar.
Resultado esperado	Se deberá de reiniciar el juego sin salir al menú principal y cambiar el orden en que se realizan las preguntas.

9.3.1. Editar

CP-009	Eliminar una ciudad de un juego ya creado
Descripción	Una vez creado un juego se podría eliminar una ciudad y modificar así las localidades que ahora se encuentran en el juego.
Acción	Ir al menú de Editar Juego, marcar una ciudad en el mapa y seleccionar eliminar.
Resultado esperado	La localidad seleccionada empezara a saltar una vez seleccionada. El icono de buscar y Guardar será sustituido por eliminar y cancelar. Aparecerá el botón de información en la esquina superior izquierda. Una vez eliminada la localidad esta deberá desaparecer del mapa.

CP-010	Añadir una ciudad a un juego ya creado
Descripción	Una vez creado un juego se podría añadir una nueva localidad al juego ya creado
Acción	Ir al menú de Editar Juego, seleccionar el icono de buscar y añadir una nueva localidad.
Resultado esperado	Aparecerá en el mapa la nueva localidad guardada junto a los otros puntos del juego seleccionado.

9.3.2.Importar Juego Actual

CP-011	Advertencia de Importar un Juego
Descripción	Tratamos de importar un juego.
Acción	Ir al menú de Importar Juego
Resultado esperado	Aparecerá un dialogo indicando las consecuencias de importar un juego.

CP-012	Importar un Juego I
Descripción	Aceptamos la importación del juego
Acción	Ir al menú de Importar Juego, aceptar la advertencia, seleccionar una dirección de almacenamiento de la cual no se tiene permisos para importar el juego como por ejemplo Drive.
Resultado esperado	Aparecerá un dialogo indicando las consecuencias que ocasionaron el fallo de importación y como corregirlos.

CP-013	Importar un Juego II
Descripción	Seleccionamos correctamente un juego a importar.
Acción	Ir al menú de Importar Juego, aceptar la advertencia, seleccionar una dirección de almacenamiento de la cual se tiene permisos como por ejemplo la memoria del teléfono o una tarjeta SD, seleccionamos importar y vamos a Jugar Juego del menú inicial.
Resultado esperado	Se mostrara en el mapa el nuevo juego cargado y este sustituirá al anterior Juego Actual.

9.3.1.Exportar Juego Actual

CP-014	Exportar Juego
Descripción	Queremos exportar el juego
Acción	Ir al menú de Exportar Juego, cerramos la advertencia, seleccionamos una aplicación que permita el envío del fichero .txt como por ejemplo el correo electrónico.
Resultado esperado	Se exportara el juego correctamente.

9.3.2.Ayuda

CP-015	Ayuda I
Descripción	El usuario inicia la aplicación por primera vez.
Acción	Instalar la aplicación.
Resultado esperado	Se mostrara la ayuda de la aplicación.

CP-016	Ayuda II
Descripción	El usuario desea ver la ayuda del juego desde el menú inicial.
Acción	Ir a la opción Ayuda del menú inicial.
Resultado esperado	Se mostrara la ayuda de la aplicación.

9.3.3.Elegir Filtro

Esta prueba se realizara para todos los filtros existentes

CP-016	Elegir filtro
Descripción	Se solicita ver los filtros disponibles
Acción	Ir a la opción Elegir Filtro y seleccionar uno. Ir a la opción Jugar del menú principal.
Resultado esperado	Se mostrara el mapa con los puntos elegidos por el filtro aplicado.

9.3.4. Modo de Juego: ¿Cómo se llama este punto?

CP-017	Jugar IV
Descripción	Se desea jugar al modo de juego: ¿Cómo se llama este punto?
Acción	Ir a la opción Modo de Juego y seleccionar ¿Cómo se llama este punto? Ir a la opción Jugar del menú principal. Introducir una respuesta.
Resultado esperado	Se mostrara el mapa donde aparecerá varios puntos a localizar. El marcador que se pregunta a qué localidad corresponde deberá de estar saltando. Cuando la respuesta es introducida se comprobara y el mapa se centrara en otro punto a localizar.

9.3.5. Niveles de juego: Intermedio, Avanzado

Los niveles de juego se probaran combinándolo para cada modo de juego y cada filtro.

CP-018	Jugar dificultad Intermedio
Descripción	Se desea jugar al nivel de juego con la dificultad intermedia.
Acción	Ir a la opción Elegir Dificultad del Juego y seleccionar: Intermedio Ir a la opción Jugar del menú principal. Introducir una respuesta.
Resultado esperado	Al seleccionar una respuesta aparecerá un dialogo indicando si la respuesta seleccionada es correcta o incorrecta. En el mapa no se mostrara si la respuesta seleccionada es correcta o incorrecta.

CP-019	Jugar dificultad Avanzado
Descripción	Se desea jugar al nivel de juego con la dificultad avanzado.
Acción	Ir a la opción Elegir Dificultad del Juego y seleccionar: Avanzado Ir a la opción Jugar del menú principal. Introducir una respuesta.
Resultado esperado	Al seleccionar una respuesta aparecerá un dialogo indicando que la respuesta esta oculta. En el mapa no se mostrara si la respuesta seleccionada es correcta o incorrecta.

9.3.1. Conexión a internet

CP-020	Comprobar conexión a Internet al crear un Juego nuevo
Descripción	Se desea crear un juego nuevo pero no se dispone de conexión a internet.
Acción	Deshabilitar el internet del dispositivo. Ir a la opción Crear Juego del menú principal.
Resultado esperado	Mostrará un error indicando que es necesario tener conexión a internet. Se mostrara la opción de reintentar.

CP-021	Comprobar conexión a Internet al intentar jugar
Descripción	Se desea jugar pero no se dispone de conexión a internet.
Acción	Deshabilitar el internet del dispositivo. Ir a la opción Jugar del menú principal.
Resultado esperado	Mostrará un error indicando que es necesario tener conexión a internet. Se mostrara la opción de reintentar.

CP-022	Comprobar conexión a Internet al intentar editar un juego
Descripción	Se desea jugar pero no se dispone de conexión a internet.
Acción	Deshabilitar el internet del dispositivo. Ir a la opción Editar Juego del menú principal.
Resultado esperado	Mostrará un error indicando que es necesario tener conexión a internet. Se mostrara la opción de reintentar.

9.4. Resultados de las pruebas realizadas

A continuación, se mostraran los resultados obtenidos para los diferentes tipos de dispositivos donde se han realizados las pruebas. Los errores encontrados serán referenciados a la tabla 19, de esta manera se pueden analizar más detalladamente.

9.4.1. Pruebas en Móvil Samsung Galaxy S3

Características:

- pantalla Super AMOLED HD 720p de 4.8 pulgadas.
- procesador Exynos 4 Quad de cuatro núcleos a 1.4GHz
- 1GB de RAM
- 16GB de memoria interna
- SO: CyanogenMod
- Versión de Android : 5.02

Resultados obtenidos:

Prueba	Versión Beta	Versión Final
CP-001	ok	ok
CP-002	Fallo(1)	ok
CP-003	ok	ok
CP-004	ok	ok
CP-005	ok	ok
CP-006	ok	ok
CP-007	ok	ok
CP-008	ok	ok
CP-009	Fallo(2)	ok
CP-010	ok	ok
CP-011	ok	ok
CP-012	ok	ok
CP-013	ok	ok
CP-014	ok	ok
CP-015	Fallo(3)	ok
CP-016	Fallo(3)	ok
CP-017	ok	ok
CP-018	ok	ok
CP-019	ok	ok
CP-020	Fallo(4)	ok
CP-021	Fallo(5)	ok
CP-022	Fallo(6)	ok

Tabla 18: Resultados de las pruebas en Samsung Galaxy S3

9.4.2. Pruebas en Samsung Galaxy Tab 3 de 7 pulgadas

Características:

- Pantalla de 7" 1024 x 600 con tecnología TFT
- Memoria RAM de 1 GB a 1000 MHz
- Procesador Intel a 1.2 GHz
- Sistema operativo: Android Jelly Bean 4.2.1
- Conexión inalámbrica WiFi GPS Bluetooth 3

Prueba	Versión Beta	Versión Final
CP-001	ok	ok
CP-002	Fallo(1)	ok
CP-003	ok	ok
CP-004	ok	ok
CP-005	ok	ok
CP-006	ok	ok
CP-007	ok	ok
CP-008	ok	ok
CP-009	Fallo(2)	ok
CP-010	ok	ok
CP-011	ok	ok
CP-012	ok	ok
CP-013	ok	ok
CP-014	ok	ok
CP-015	Fallo(3) Fallo(7)	ok
CP-016	Fallo(3) Fallo(7)	ok
CP-017	ok	ok
CP-018	ok	ok
CP-019	ok	ok
CP-020	Fallo(4)	ok
CP-021	Fallo(5)	ok
CP-022	Fallo(6)	ok

Tabla 19: Resultados de las pruebas en Samsung Galaxy Tab 3

9.4.1. Fallos encontrados

A continuación, se detallan los fallos encontrados así como sus posibles soluciones.

Fallo	Descripción	Posible Solución
1	Al no encontrar los datos a mostrar en el botón de información el campo se rellenaba como “null”. Por ejemplo, si introducíamos España y dábamos al botón de información los campos de provincia y comunidad salían como “null.”.	Agregar una condición donde si el campo de la localidad es “null”, este será sustituido por “No disponible”.
2	Al seleccionar una localidad a eliminar no salía el dialogo de confirmación.	Desarrollar el dialogo de confirmación ya que este no está creado. Se creara para ser usado genéricamente en cualquier idioma, solo habría que añadir su correspondiente traducción en el fichero strings.xml de dicho idioma.
3	La ayuda no estaba configurada para el idioma Ingles.	Como la estructura esta creada, basta con añadir las imágenes del tutorial en ingles en la carpeta “drawable-en-hdpi”.
4	La conexión a internet se comprobaba cuando se iba a buscar una ciudad y no cuando se estaba cargando el mapa, por lo cual si no había conexión este no cargaba.	Añadir la comprobación realizada para buscar una localidad al inicio de la actividad, además de la opción reintentar.
5	No se comprobaba la conexión a internet y el mapa no cargaba si esta no existía.	Añadir la comprobación de conexión a internet.
6	No se comprobaba la conexión a internet y el mapa no cargaba si esta no existía.	Añadir la comprobación de conexión a internet.
7	La ayuda no se visualiza correctamente en el dispositivo debido al tamaño de la pantalla.	Configurar correctamente la vista para que se adapte a la pantalla de 7 pulgadas.

Tabla 20: Fallos encontrados

Capítulo 10

Conclusiones

10.1. Objetivos Cumplidos

- Se ha realizado un estudio para la realización de proyectos Android, además de cómo desarrollar este en el entorno de desarrollo Eclipse usando el SDK de Android.
- Se ha conseguido vincular la API v3 de Google Maps JavaScript a un proyecto Android.
- Se ha conseguido una comunicación correcta para que la captura y el tratamiento de eventos se realicen tanto en Android como en la página HTML.
- Se ha conseguido una aplicación lo más nativa posible a pesar de que el mapa es cargado en una página web.
- Se ha logrado obtener datos de una localidad como por ejemplo, sus coordenadas, dado solo su nombre.
- Se ha conseguido una interfaz bastante intuitiva y atractiva.
- Se ha conseguido importar y exportar un juego usando para la búsqueda del directorio un gestor de archivos ya creado.
- Se puede Jugar a un juego ya creado o a uno importado de una manera correcta.
- Se muestran varios niveles de juego (Principiante, Intermedio, Avanzado), cada uno con su peculiaridad.
- Se consiguió realizar un filtro de las localidades ya existentes en la base de datos. De esta manera a pesar de que el juego en un principio fue diseñado para que solo se pudiera jugar a un Juego Actual, podemos reutilizar localidades creadas en juegos anteriores.
- Se realizaron dos modalidades de Tipo de Juego: ¿Dónde está X? y ¿Cómo se llama este punto?
- Se ha conseguido realizar una aplicación lo más flexible posible, de esta manera es posible añadir más idiomas y filtros en un futuro.
- Actualmente la aplicación está desarrollada para el idioma Español e Inglés.
- La adaptación de la aplicación para varios tipos de pantalla es bastante correcta, teniendo en cuenta que Android cuenta con una variedad muy grande.
- No se ha necesitado guardar ningún dato personal, consiguiendo la privacidad total de los posibles usuarios.

10.2. Valoración Personal

La realización de este TFG me ha permitido adquirir nuevos e interesantes conocimientos al tiempo que iba afianzando los que ya había obtenido durante mi formación como Ingeniero Informático.

En las distintas etapas del desarrollo de un proyecto de software puede que la tecnología inicialmente pensada a usar no sea la correcta o esta no se adapte a las necesidades del proyecto. La capacidad para adaptarme a estos aspectos ha resultado clave para superar estos escollos y llevar el proyecto por el camino adecuado.

Las aplicaciones móviles, en mi opinión, son el presente y el futuro de los sistemas informáticos y están en pleno auge. Podemos encontrar un SO Android en muchos tipos de dispositivos como relojes, móviles o coches. Todo esto unido a que es un lenguaje de programación libre y a su facilidad para comerciar una app en su plataforma PlayStore, fueron factores clave para despertar la idea de realizar el TFG en Android.

En los últimos tiempos se ha convertido en algo muy habitual la programación no nativa usando plataformas de desarrollo que te permiten desplegar un programa en varios tipos de sistemas. Este tipo de programación favorece la reutilización del código, repercutiendo en un mayor ahorro de coste y un menor tiempo de desarrollo de una app, ya que con un único código podemos lanzarla a varias plataformas. En mi opinión, y sin quitarle importancia a lo anterior dicho, creo que para el desarrollo de aplicaciones de calidad sigue siendo muy importante conocer el lenguaje nativo y programar en este, por eso el TFG se ha desarrollado usando el entorno de desarrollo de Eclipse y el SDK de Android.

10.3. Trabajo Futuro

- Mejorar la interfaz gráfica, añadiéndole animaciones y sonido.
- Mejorar la exportación e importación de los juegos creados, presentando una mayor accesibilidad de las aplicaciones y de los directorios del dispositivo.
- Añadir una opción que permita ver las estadísticas del jugador en una interfaz sencilla, atractiva y útil.
- Crear la opción de multijugador, de manera que se pueda compartir un mismo mapa entre varios jugadores.

Bibliografía

Libros

[1] Universidad Complutense de Madrid. Introducción a Android. E.M.E Editorial. ISBN: 978-84-96285-39-5.

Sitios web

[2] Developer Android: <http://developer.android.com/>

Descripción: Página oficial para desarrolladores de Android

[Última visita: 12-08-2015]

[3] Google Developers: <https://developers.google.com>

Descripción: Página para desarrolladores de Google. Se utilizó para ver todo lo relacionado con el uso del mapa de la API v3.

[Última visita: 2-08-2015]

[4] Una docena de...: <http://unadocenade.com/una-docena-de-alternativas-a-google-maps/>

Descripción: Se utilizó para ver las distintas alternativas para el uso de mapas.

[Última visita: 10-05-2015]

[5] Wikipedia: <https://es.wikipedia.org>

Descripción: Enciclopedia interactiva que se utilizó para ver conceptos y definiciones en las distintas etapas del proyecto.

[Última visita: 17-08-2015]

Bibliografía

[6] Open Layers: <http://openlayers.bicimap.es/manualOpenLayers.html>

Descripción: Manual de Open Layers que se utilizó cuando se buscaban alternativas para el uso de mapas en la app.

[Última visita: 12-05-2015]

[7] MapsQuest: <https://developer.mapquest.com/solutions>

Descripción: Página de desarrolladores de MapsQuest que se utilizó mientras se miraban las alternativas a los mapas para el desarrollo de la app.

[Última visita: 12-05-2015]

[8] Curso de Programación Android: <http://www.sgoliver.net/blog/>

Descripción: Blog con diferentes tutoriales de Android que ha servido para desarrollar la etapa de implementación.

[Última visita: 17-07-2015]

[9] Curso de Programación Android: <http://stackoverflow.com/>

Descripción: Sitio web para la comunidad de desarrolladores informáticos en el cual se ha encontrado soluciones a problemas de implementación.

[Última visita: 12-08-2015]

[10] Monografias.com: <http://www.monografias.com/trabajos101/sistema-operativo-android/sistema-operativo-android.shtml>

Descripción: Centro de tesis, documentos, publicaciones y recursos educativos

[Última visita: 10-07-2015]

[12] La Vanguardia: <http://www.lavanguardia.com/vangdata/20150409/54429740418/android-domina-mercado-smartphones-en-espana-con-87-6-ventas.html>

Descripción: Sitio web del periódico La Vanguardia.

[Última visita: 12-08-2015]

[13] GeekyTheory: <https://geekytheory.com/>

Descripción: Sitio web donde se pueden encontrar tutoriales de Android.

[Última visita: 17-08-2015]

[14] El Androide libre: <http://www.elandroidelibre.com/>

Descripción: Sitio web con documentación sobre Android

[Última visita: 15-08-2015]

[15]Android Curso: <http://www.androidcurso.com/>

Descripción: Diploma de especialización en desarrollo de aplicaciones Android de la Universidad Politécnica de Valencia [Última visita: 12-08-2015]

[16]UPEDu: <http://upedu.org/>

Descripción: Sitio oficial del Proceso Unificado aligerado para educación.

[Última visita: 18-08-2015]

[17] V-Mann: <http://www.v-mann.com/>

Descripción: Consultoría y desarrollo TIC

[Última visita: 13-08-2015]

[18]Microsoft: <https://msdn.microsoft.com/>

Descripción: Página de Microsoft orientada a desarrolladores

[Última visita: 8-06-2015]

Blog

[11]Azone: <http://androidzone.es>

Descripción: Es un blog de Android en español.

[Última visita: 13-06-2015]

[19]Juan Minaya León: <http://blog.juanminaya.com/>

Descripción: Blog con la descripción de distintos patrones de diseño.

[Última visita: 8-06-2015]

[20]roweb: <http://www.roweb.ro/>

Descripción: Blog para desarrolladores

[Última visita: 9-06-2015]

Apéndice A: Manual de Usuario

Instalación de Geolocalization Game

1. Vaya a la tienda de aplicaciones (PlayStore) de su dispositivo Android.
2. Introduzca: Geolocalization Game.
3. Presione: instalar
4. Acepte los permisos que necesita la aplicación para su correcto funcionamiento.

Primera Vez

La primera vez que ejecute la aplicación habrá algunas funciones deshabilitadas, esto nos permitirá guiarte un poco en el proceso de inicio de la aplicación. Una vez creado o importado un juego estas funciones serán habilitadas.



Figura 27: Pantalla de Menú Principal de la primera instalación.

Crear un Juego

Botón Buscar: Permite la búsqueda de cualquier localidad. Aunque en principio esta aplicación está pensada para el aprendizaje geográfico de localidades como por ejemplo pueblos, ciudades, comunidades y países, el servicio de búsqueda abarca más allá. Geolocalization Game te permite localizar cualquier sitio como por ejemplo el Parque Nacional Yellowstone, una calle o barrio.

¿Qué hacer si no sale la localidad que quiero cuando la busco?: A veces puede ocurrir que la localidad a buscar tenga el mismo nombre en varios sitios y el buscador elija la que no esperamos nosotros. Lo que tenemos que hacer en este caso es especificar la localidad, por ejemplo, en el caso de buscar una ciudad, además del nombre de esta escribiríamos el país en el que se encuentra.

Botón guardar: Como su nombre indica, este botón guarda la localidad encontrada y la añade al juego que estamos creando.

Botón Salir: En caso de no haber guardado ninguna localidad, no se crea ningún juego nuevo. Por el contrario si hemos guardado una localidad nueva o más de una, se crea un juego con dichos puntos a localizar. **Importante:** El juego creado sustituirá al ya existente en caso de haberlo. Las localidades de juegos anteriores no se pierden, sino que son almacenadas para futuros juegos configurables en la opción de “Elegir Filtro”.

Botón Información: Te permite ver datos como el nombre, provincia, comunidad y país de la localidad buscada, siempre que esta información esté disponible.

Mapa: Permite mostrar el relieve del mapa mostrado.

Satélite: Muestra una vista satelizar del mapa y permite hacer zoom sobre la pantalla usando los dos dedos.

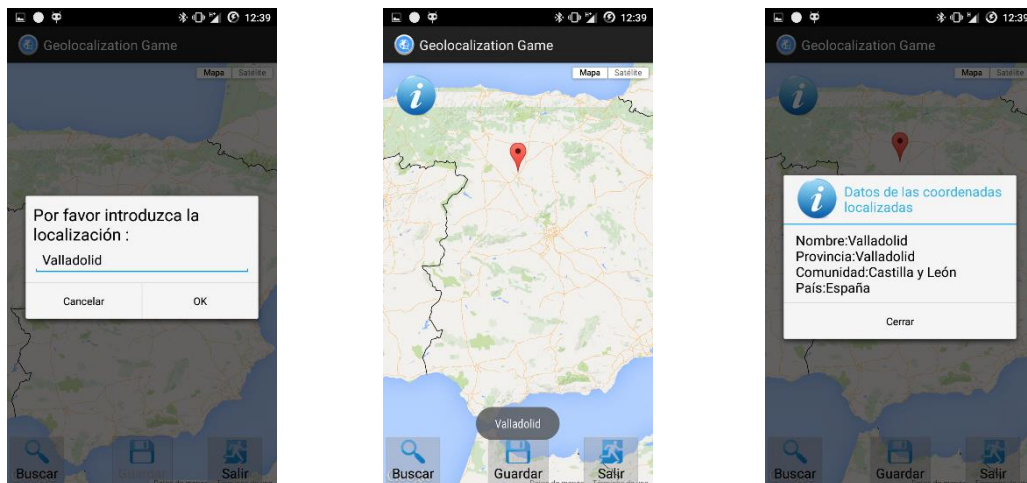


Figura 28: Crear Juego

Jugar

La configuración inicial del juego es la siguiente:

Elegir Filtro: Juego Actual (juego creado o importado).

Elegir Modo de Juego: ¿Dónde está X?

Elegir Dificultad del Juego: Principiante.

En la esquina superior izquierda del juego, junto al logo de la aplicación, irán apareciendo una serie de preguntas donde se te pida localizar aleatoriamente los puntos añadidos en el juego q has creado.

1. Selecciona con el dedo la respuesta que creas correcta.
2. Aparecerá un mensaje indicándote el resultado de lo que has seleccionado (niveles principiante e intermedios).
3. En el mapa aparecerá marcado este resultado (nivel principiante).
4. Una respuesta incorrecta puede volver a ser marcada como correcta en caso de que se diera esta situación.
5. Al final del juego aparecerá la puntuación alcanzada y los nombres de las localidades en que has fallado.

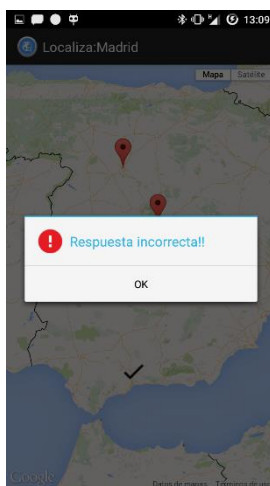


Figura 29: Jugar

Editar un Juego

La edición de juego te permite añadir o eliminar localidades del juego al que estás jugando en ese momento. De este modo no es necesario crear un juego nuevo si lo que quieres es modificar solo unos puntos del ya existente.

Botón buscar: Te permite añadir una localización nueva. Puede ver más información sobre este botón en la sección Crear Juego.

Botón Salir: Sales de la opción Editar Juego. **Importante:** Cuando se elimina o se añade una localización se guardan estas modificaciones, no permitiendo echar atrás el cambio hecho. En el caso de que eliminemos algo erróneamente siempre podemos volverla añadir.

Botón Eliminar: Si seleccionas una localización en el mapa, se habilitara la opción de “Eliminar”. Antes de eliminar definitivamente un punto aparecerá una confirmación, una vez aceptada esta, no se podrá deshacer el cambio.

Botón Cancelar: Permite desmarcar una localidad seleccionada.

Botón Información: Muestra la información de la localidad seleccionada en caso de esta existir.

El juego que se edita esta correspondido con la opción de filtro que tengamos seleccionado. De esta manera si tenemos seleccionada la opción “Sin Filtro” y eliminamos una localidad que se encuentra en el juego actual, esta será elimina de ahí también.

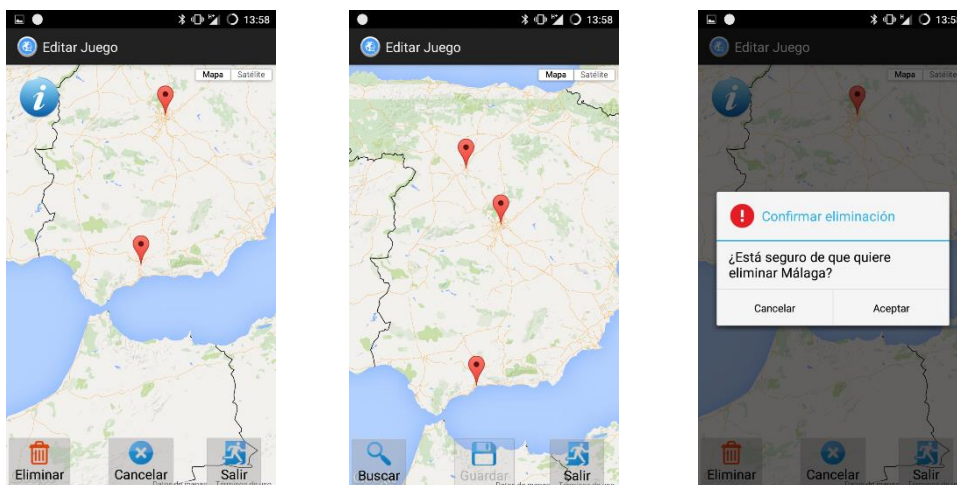


Figura 30: Editar Juego

Importar Juego Actual

La importación de un juego te permite jugar a Geolocalization Game sin necesidad de tener que crear un Juego. El gestor de fichero para navegar por el dispositivo en busca del fichero a importar depende de cada dispositivo. En caso de no poder encontrar el fichero del juego, te recomendamos que descargues un gestor de fichero nuevo en el PlayStore. **Importante:** cuando se importa un juego se sustituye el Juego Actual, aunque estos datos seguirán existiendo para otras modalidades de la opción Elegir Filtro. En caso de que no quieras perder el Juego Actual te recomendamos que antes de importar exportes el juego, de esta manera siempre puedes volver a cargar dicho juego.

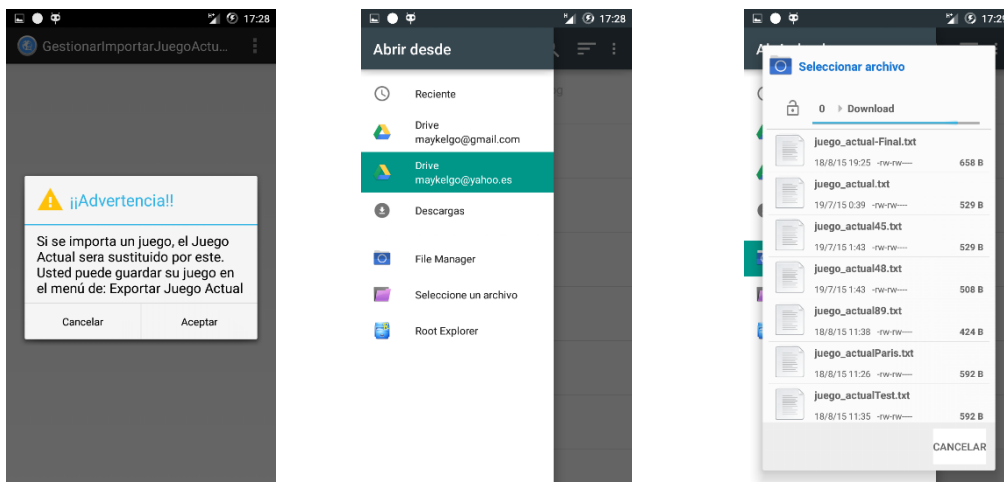


Figura 31: Importar Juego

Exportar Juego Actual

La exportación de un juego te permite poder compartir tu Juego Actual con otra persona o guardarlo para otro momento. El sistema buscara en el dispositivo posibles aplicaciones que puedan compartir el fichero con el juego. Puede que algunas aplicaciones no funcionen correctamente al no permitir estas el envío de ficheros. En todo caso, nosotros te recomendamos que para la exportación de un fichero utilices alguno de gestor de correo o Bluetooth. **Importante:** WhatsApp no permite el envío de ficheros con la extensión .txt. En futuras versiones esto será mejorado, de manera que se consiga una exportación más fácil y menos confusa.

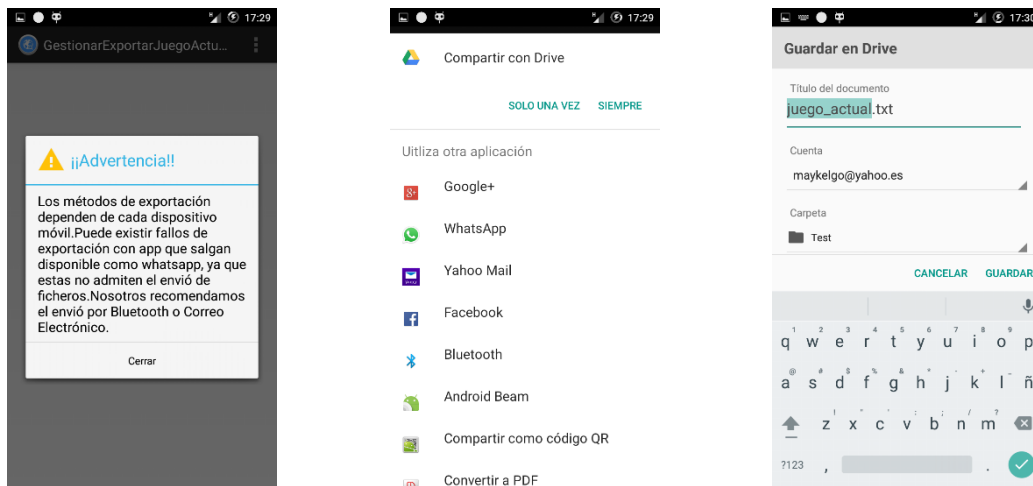


Figura 32: Exportar Juego

Ayuda

La Ayuda de la aplicación será mostrada la primera vez que inicies la aplicación con el objetivo de que te resulte más fácil el uso de la app. Siempre que quieras podrás acceder desde este menú a la ayuda del juego.

Botón Finalizar tutorial: Te permitirá salir en cualquier momento del tutorial.



Figura 33: Ayuda

Opciones

Geolocalization Game cuenta con un conjunto de opciones que podrás configurar para que mejores la experiencia y dificultad del juego.

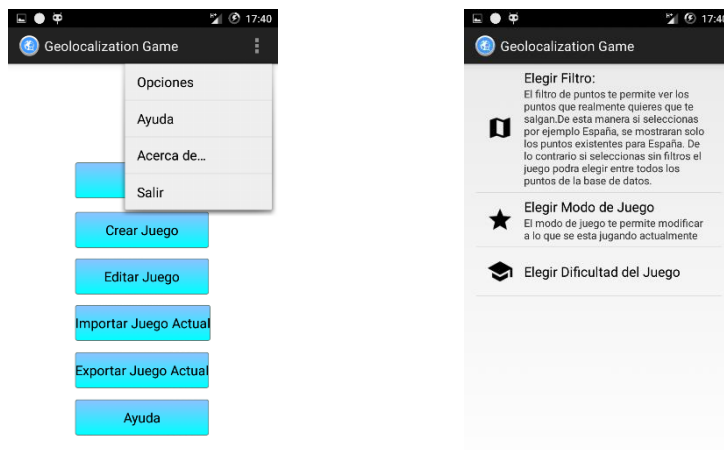


Figura 34: Opciones

Elegir Filtro

- **Juego Actual:** Esta opción es la que trae aplicada el juego por defecto y está asociada al último juego que has creado o importado.
- **Sin Filtro:** Seleccionaras todos las localidades que has ido añadiendo a medida que has ido creando, editando o importando un juego.
- **España:** Se parece a la opción Sin Filtro, solo que aquí seleccionaríamos solo los que corresponden a España.
- **Francia:** Igual que la opción de España pero esta vez seleccionaremos los de Francia.

Modo de Juego

- **¿Dónde está X?:** Tendrás que seleccionar un punto en el mapa dado el nombre de una localidad. Para más información ver la sección de Jugar.
- **¿Cómo se llama este punto?:** El sistema te preguntara por el nombre de una localidad, esta aparece con una animación de salto y centrada en el mapa. Deberás de introducir el nombre de esta localidad. No se tendrá en cuenta las mayúsculas para su corrección, pero si las tildes.

Elegir Dificultad del Juego

- **Principiante:** Opción seleccionada por defecto. Esta dificultad te permitirá ver si la respuesta introducida es correcta o no. Para que te sea más fácil el sistema te lo indicara cuando introduzcas la respuesta y te la dejara marcada en el mapa. Para más información ver la sección de Jugar.
- **Intermedio:** El sistema te mostrara el resultado de tu respuesta (correcta o incorrecta), pero no la dejara marcada en el mapa. Esta dificultad te permitirá entre otras cosas ganar en memoria.
- **Avanzado:** No te indicara si es correcta o incorrecta ni tampoco se te quedara marcado en el mapa. Podrás ver los nombres de las localidades falladas una vez finalizada la partida.

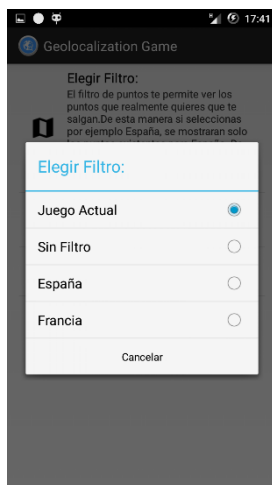


Figura 35: Elegir Filtro

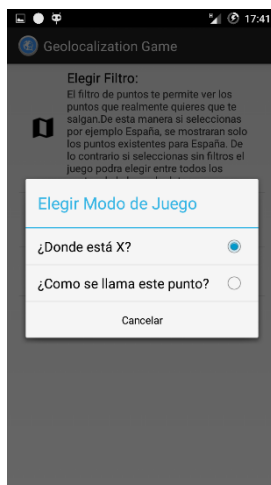


Figura 36: Modo de Juego

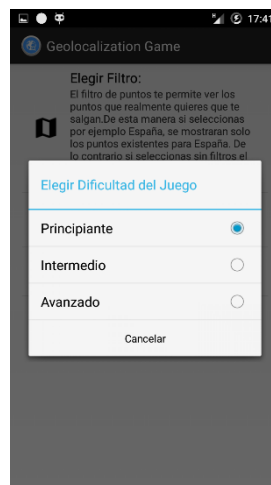


Figura 37: Dificultad del Juego

Apéndice B: Glosario de términos

- TFG : Trabajo de fin de grado
- OSM: OpenStreetMap
- SO: Sistema Operativo
- OHA: Open Handset Alliance
- APIs : interfaz de programación de aplicaciones
- URL: Uniform Resource Identifier(identificador uniforme de recurso)
- RUP: Proceso Racional Unificado
- MVC: modelo-vista-controlador
- MVP: modelo-vista-presentador

Apéndice C: Contenido del CD

Memoria del TFG

Aquí se encontrarán la memoria del proyecto (memoria.pdf), y el fichero que contendrá todos los diagramas realizados (diagramas.ast)

Software

Contendrá todo el código de la aplicación desarrollada.

Instalador

Esta carpeta contiene el instalador de la aplicación (GeolocalizationGame.apk).

Manuales

Manual de instalación y de usuario

Imágenes utilizadas

Todas las imágenes que se han utilizado o se han editado durante el TFG